



CogTool User Guide

Version 1.2

May 23, 2012

Bonnie E. John

IBM T. J. Watson Research Center
Software Productivity
19 Skyline Drive, Hawthorne NY 10532

<http://cogtool.org>

Addendum for CogTool Version 1.2

CogTool 1.2 is essentially the same as CogTool 1.1 for making storyboards and predicting skilled performance time. Therefore, this User Guide presents the new features of CogTool 1.2 here and then the User Guide for version 1.1 follows in its entirety.

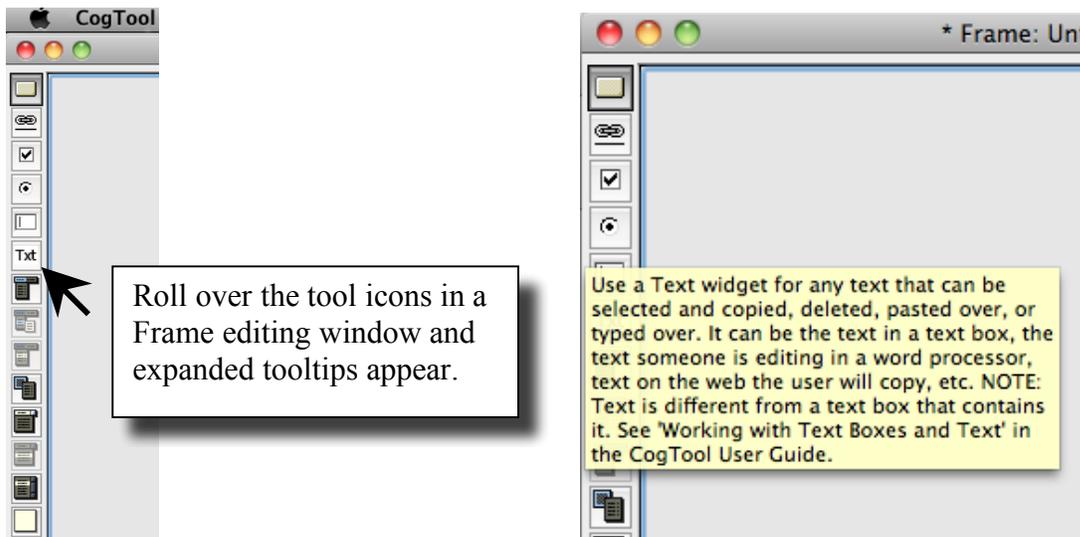
New platforms for CogTool 1.2

CogTool 1.2 now runs in Windows 7, Mac OS 10.6 (Snow Leopard), and Mac OS 10.7 (Lion).

Improvements to the GUI in CogTool 1.2

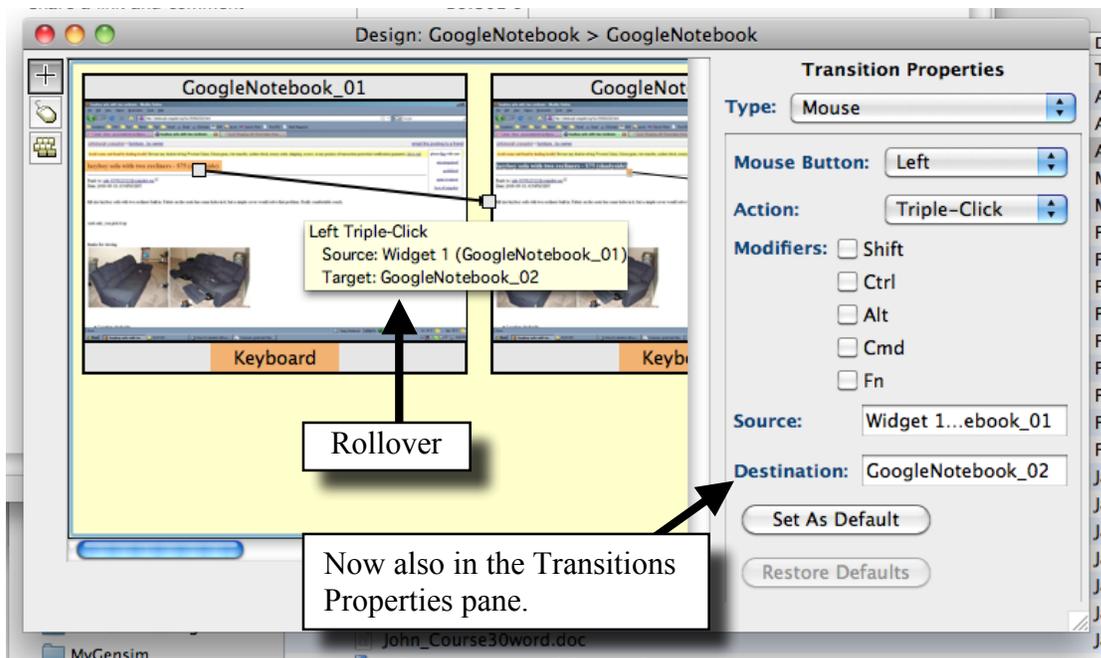
Widget Tool Tips

In response to user requests, we have expanded the tool tips on the Widget Toolbar in Frame Editing windows. The tooltips now contain the prose in **Appendix B: Types of Widgets and When to Use Them** (but continue to refer to that appendix if you want to see pictures of example widgets).



Transition Properties Pane now contains source & destination information

Formerly only available by rolling over a transition in a Design window, the source and destination of a transition are now displayed in the transition's property pane as well.



Changed or new functionality in CogTool 1.2

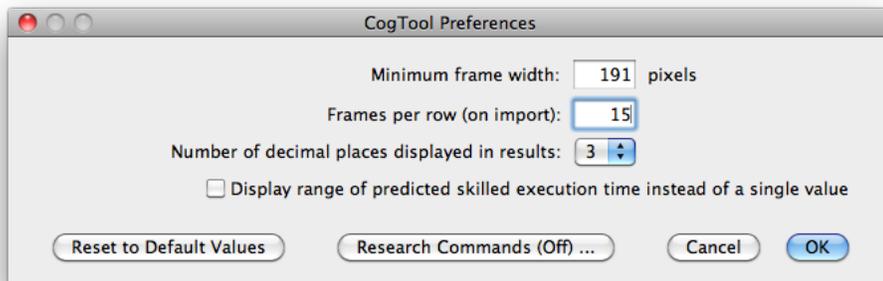
Default starting position of the hands

Formerly defaulting to starting on the keyboard, our users told us that most of their tasks start with the user's hand on the mouse. CogTool 1.2 now defaults to starting with the hand on the mouse, if a mouse is one of the available devices.

(This will not effect the initial starting position of the hands in existing models made with versions prior to 1.2.)

Numeric results in the Project Window can now be changed in the Preferences dialog

Prior to v1.2, the Project window always displayed the predictions of skilled task execution time in seconds with three decimal places. Users of CogTool found this misleading because the predictions are known to be within $\pm 10\%$ of what you would find if you could collect data of skilled users performing these tasks. To make communicating the results more plausible, we have added two adjustments in the Preferences dialog box.

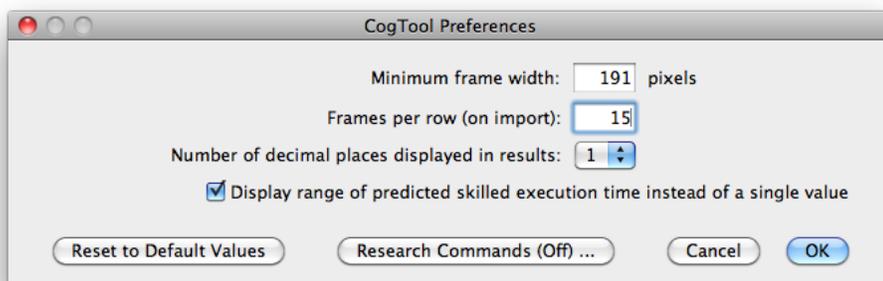


Setting the number of decimal places to 3 and not checking the “Display range” box retains the Project window as it appeared prior to version 1.2, as shown below.

Project: CollaborativeShopping

Tasks	GoogleNotebook	Gmail	Wetpaint
Share link and comment	17.673 s	33.600 s	33.069 s

But you can now change this presentation of results. For example, setting the number of decimal places to 1 and checking the “Display range” box highlights the fact that GoogleNotebook is faster than the other two designs for this task, but Gmail and Wetpaint are not sufficiently different to be detected by CogTool models.



Project: CollaborativeShopping

Tasks	GoogleNotebook	Gmail	Wetpaint
Share link and comment	17.7 (15.9, 19.4) s	33.6 (30.2, 37.0) s	33.1 (29.8, 36.4) s

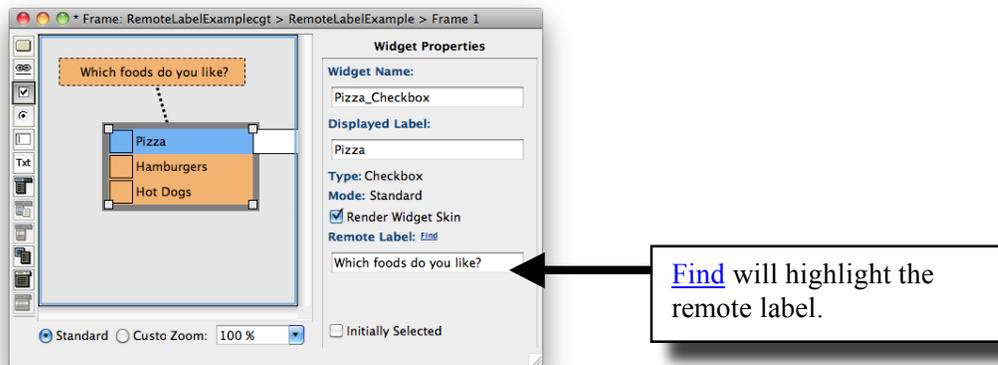
You can reset all parameters to CogTool’s default values (displaying 1 decimal place, but no range) by clicking the new “Reset to Default Values” button.

Modeling dragging on touch screen devices.

You can now model dragging on touch screen devices with transitions called “Up Tap” and “Down Tap.” Just use a Down-Tap self-transition on the widget where the drag starts and an Up-Tap transition on a widget where the drag ends. (See this explained for dragging with a mouse in Section 4.4.6.3 Dragging Over Text to Highlight It.)

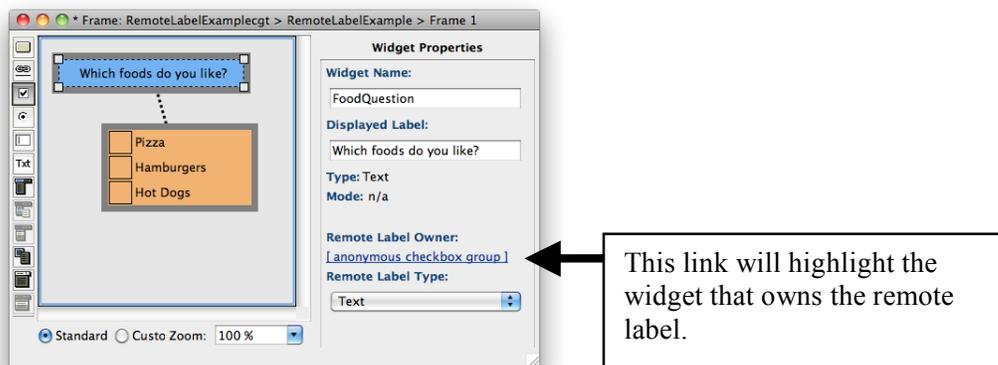
Widgets can now have a remote label

Most widgets can now have what we call a “remote label” which will make using CogTool as a prototyping tool more realistic. For example, a group of check boxes often has some text asking a question that the user can answer by checking one or more of the checkboxes. This text is represented as a remote label of a standard checkbox widget, as shown below. The remote label is connected to the widget (or widget group) that owns it with a dashed line.



The remote label is itself a widget of Type Text, so you can model interacting with it as you would any other text in an interface.

If your interface is complex, it is possible that the remote label might be quite far from the widget that owns it, so when a widget is selected, you can find its remote label by clicking on the [Find](#) link, which will select the remote label (below). Likewise, when a remote label is selected, you can always find its owner by clicking on the link below “Remote Label Owner:”





CogTool User Guide

Version 1.1

17 September 2009

Bonnie E. John, Principle Investigator

Human-Computer Interaction Institute

School of Computer Science

Carnegie Mellon University

<http://cogtool.org>

© 2008, 2009 Bonnie E. John and Carnegie Mellon University

ACKNOWLEDGMENTS

The following people have contributed to CogTool. We would like to thank them for their hard work and patience with the process.

Gus Prevas, Ken Koedinger, Peter Centgraf, Mike Horowitz, Alex Eiser, Alex Faaborg, Sandy Esch, Jason Cornwell, Lily Cho, Don Morrison, Samantha Konwinski, Carmen Jackson, Josh Ehlke, Ryan Myers, Diana Dill, Leigh Johnston, Chris Monti, Melissa Gallagher, Annie Luo, Brett Harris, Khaled Ziyaeen, Alonso Vera, Collin Green, Guy Pryzak, Mike Feary, Rick Lewis, Mason Smith, Andrew Howes, Peter Pirolli, Wai-Tat Fu, Victoria Bellotti, Dario Salvucci, Lance Sherry, Maricel Medina Mora, Marilyn Hughes Blackmon, Peter Polson, Karl Fennel, and all the students in HCI Methods and Cognitive Crash Dummies courses at Carnegie Mellon, Stanford and professional conferences.

This research was supported by funds from the Office of Naval Research, N00014-03-1-0086, NASA Ames Research Center, DSO National Laboratories, the Boeing Corp., NEC, and IBM. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, NASA, DSO National Laboratories, Boeing, NEC, IBM or the U. S. Government.

CONTENTS

CONVENTIONS FOR THIS USER GUIDE	1
CHAPTER 1: OVERVIEW	2
1.1 What is CogTool?.....	2
1.2 How CogTool Works	2
1.3 Uses of CogTool.....	4
CHAPTER 2: GETTING STARTED WITH COGTOOL	6
2.1 Downloading CogTool.....	6
2.2 Installing CogTool	6
2.2.1 Installing on Mac OS X.....	6
2.2.2 Installing on Windows XP and Vista.....	6
2.3 Launching CogTool.....	7
2.4 Exploring CogTool.....	8
2.5 CogTool Help.....	8
CHAPTER 3: MANAGING A PROJECT	11
3.1 What is a Project?	11
3.2 The Project Window	11
3.3 Creating a New Project.....	12
3.4 Opening an Existing Project.....	13
3.5 Saving a Project.....	13
3.6 Closing a Project.....	14
3.7 Project Properties	14
3.8 Reopening Project Windows and Understanding the Windowing System.....	15
CHAPTER 4: PROTOTYPING AN INTERFACE	17
4.1 Designs	17
4.1.1 What is a Design?.....	17
4.1.2 What is a Device?.....	17
4.1.3 The Design Window	18
4.1.4 Creating a New Design	19
4.1.5 Selecting a Design.....	20
4.1.6 Editing a Design	20
4.1.7 Saving a Design	21
4.1.8 Closing a Design	21
4.1.9 Moving a Design.....	21
4.1.10 Cutting, Copying, and Pasting a Design.....	22
4.1.11 Deleting a Design	23
4.1.12 Duplicating a Design.....	23
4.1.13 Renaming a Design	23
4.1.14 Bringing a Design Window to the Foreground.....	24
4.1.15 Zooming a Design.....	24
4.1.16 Adding Devices to a Design	25
4.2 Frames	25
4.2.1 What is a Frame?.....	25
4.2.2 The Frame Window.....	26

4.2.3	Setting the Default size of a Frame	27
4.2.4	Creating a New Frame	27
4.2.4.1	Creating new Frames from a folder of images	28
4.2.4.2	Creating new Frames containing a “template” of Widgets.....	29
4.2.5	Selecting Frames.....	30
4.2.6	Renaming a Frame	30
4.2.7	Moving and Aligning Frames.....	31
4.2.8	Cutting, Copying, and Pasting a Frame	31
4.2.9	Deleting a Frame.....	32
4.2.10	Duplicating a Frame	32
4.2.11	Editing a Frame.....	33
4.2.12	Zooming a Frame	33
4.2.13	Setting the Background Image of a Frame	33
4.2.14	Removing the Background Image of a Frame.....	34
4.2.15	Bringing a Frame to the Foreground.....	34
4.2.16	Closing the Frame Window	34
4.3	Widgets.....	35
4.3.1	What is a Widget?	35
4.3.2	Types of Widgets	36
4.3.2.1	Standard or Custom Mode?	36
4.3.2.2	Choosing Widget Types	37
4.3.2.3	A Road map of the Widgets Section.....	37
4.3.3	Common Actions on Widgets	37
4.3.3.1	Creating a New Widget.....	37
4.3.3.2	Naming a Widget and Entering its Display Label	38
4.3.3.3	Selecting Widgets	40
4.3.3.4	Setting a Widget’s Appearance.....	40
4.3.3.4.1	Setting a Widget Layer Color	40
4.3.3.4.2	Rendering a Widget Skin.....	40
4.3.3.4.3	Setting and Removing an Image for a Widget.....	41
4.3.3.5	Moving and Aligning Widgets	42
4.3.3.6	Layering Widgets.....	43
4.3.3.7	Resizing Widgets	43
4.3.3.8	Cutting, Copying and Pasting Widgets	43
4.3.3.9	Deleting Widgets.....	44
4.3.3.10	Duplicating a Widget.....	44
4.3.3.11	Creating a “Template” of Widgets	45
4.3.4	Simple Widgets: Links, Graffiti®, and Non-Interactive.....	46
4.3.5	Moded Widgets: Buttons and Checkboxes	47
4.3.5.1	Buttons	47
4.3.5.2	Checkboxes	48
4.3.5.2.1	Creating Checkbox Widgets.....	49
4.3.5.2.2	Arranging Checkbox Widgets	51
4.3.6	Radio Buttons: A Set of Widgets Working Together.....	52
4.3.6.1	Creating Radio Button Widgets.....	52
4.3.6.2	Arranging Radio Buttons.....	54
4.3.7	List Boxes: Widgets Combined to Make a Whole	54

4.3.7.1	Creating a List Box	54
4.3.7.2	Rearranging List Box Items.....	55
4.3.7.3	Moving List Box Widgets	56
4.3.8	Pull-Down Lists: More Complex Behavior	56
4.3.8.1	Creating Pull-Down List Widgets	56
4.3.8.2	Rearranging Pull-Down List Items	58
4.3.8.3	Moving Pull-Down List Widgets.....	59
4.3.9	Menus and Context Menus: Highly Complex Behavior	59
4.3.9.1	Creating a Menu System	59
4.3.9.1.1	Creating Menus that are Always Visible	59
4.3.9.1.2	Creating a Context Menus.....	63
4.3.9.2	Selecting Menu Widgets.....	64
4.3.9.3	Re-sizing Menu Widgets	66
4.3.9.4	Moving Menu Widgets.....	66
4.3.9.5	Copying/Cutting/Pasting Menu Headers and Their Items	67
4.3.9.6	Copying/Cutting/Pasting an Entire Menu System	67
4.3.10	Working with Text Boxes & Text.....	68
4.3.10.1	Creating a Text Box.....	69
4.3.10.2	Creating Editable Text, an Example.....	70
4.3.10.2.1	Place the Cursor, Delete and Type the Right Letters.	71
4.3.10.2.2	Double-Click on the Incorrect Name, Type the Correct Name.	71
4.3.10.2.3	Drag to Highlight the Incorrect Name, Type the Correct Name.	72
4.4	Transitions.....	73
4.4.1	What is a Transition?	73
4.4.2	Creating New Transitions	74
4.4.2.1	Transitions from a Simple Widget.....	74
4.4.2.2	Transitions from an Interactive Widget: Standard Menus, Context Menus, and Pull-Down Lists.....	76
4.4.2.3	Transitions from the Keyboard or Microphone	77
4.4.2.4	Transitions from a Graffiti® Widget.....	79
4.4.3	Prototyping a System Delay	79
4.4.4	Modifying a Transition.....	80
4.4.4.1	Changing the Properties of a Transition.....	80
4.4.4.2	Changing the Source and Destination of a Transition	80
4.4.5	Deleting a Transition	81
4.4.6	Prototyping Some Common User Actions.....	81
4.4.6.1	A Series of Clicks/Taps on the Same Button (e.g., Buttons that Navigate or Scroll)	81
4.4.6.2	Drag-and-drop.....	82
4.4.6.3	Dragging Over Text to Highlight it	83
4.5	Working with Sound.....	84
4.5.1	Microphone.....	85
4.5.2	Speaker.....	85
4.6	Working with HTML	86
4.6.1	Import a Design from HTML.....	86
4.6.1.1	Known problems when importing from HTML.....	90
4.6.2	Export a Design to HTML	91
CHAPTER 5:	QUANTITATIVE ANALYSIS	95

5.1 Tasks and Task Groups.....	95
5.1.1 What are Tasks and Task Groups?.....	95
5.1.2 Creating a New Task	96
5.1.3 Creating a New Task Group	96
5.1.4 Selecting Tasks or Task Groups	97
5.1.5 Cutting, Copying, and Pasting Tasks or Task Groups	97
5.1.6 Duplicating Tasks or Task Groups	98
5.1.7 Renaming a Task or Task Group.....	98
5.1.8 Deleting Tasks or Task Groups	99
5.1.9 Changing the Order of Tasks or Task Groups	99
5.1.10 Changing the Display Value of a Task Group	99
5.2 Demonstrating How to Do a Task	100
5.2.1 What are Demonstrations and Scripts?	100
5.2.2 The Script Windows.....	101
5.2.2.1 The Select the Start Frame Window	101
5.2.2.2 The Script Edit Window.....	102
5.2.3 Opening a Script for Editing.....	103
5.2.4 Selecting a Start Frame for the Script	103
5.2.5 Demonstrating Actions for a Script.....	104
5.2.5.1 Simple Demonstration	104
5.2.5.2 Here's Why Using the Right Widgets was Important	105
5.2.5.3 Adding Self-Transitions During Demonstration	105
5.2.5.4 Adding Look-At Steps During Demonstration	106
5.2.5.5 Adding or Removing Think Steps During Demonstration.....	106
5.2.5.6 Examples: Demonstrating the Three Text-Editing Tasks Designed in Previous Sections .	108
5.2.6 Editing Think Script Steps	111
5.2.7 Deleting Script Steps	111
5.2.8 Reviewing Your Scripts	112
5.2.8.1 Reviewing the Steps in a Task Script	112
5.2.8.2 Reviewing the Steps in a Summed Task Group.....	112
5.3 Computing a Prediction.....	112
5.3.1 Recomputing Scripts	113
5.4 Exporting Scripts and Results to CSV Files	114
5.4.1 Exporting a Script to CSV.....	114
5.4.2 Exporting All Results to CSV	115
5.5 Model Visualization.....	116
REFERENCES	120
APPENDIX A: DEVICES	122
Input Devices.....	122
Output Devices	124
APPENDIX B: TYPES OF WIDGETS AND WHEN TO USE THEM	126
Button	126
Link.....	127
Checkbox.....	128
Radio Button.....	129
Text Box.....	130
Text.....	130

Menu	131
Submenu (Necessary inCustom only)	132
Menu Item(Necessary inCustom only).....	132
Context Menu	133
Pull-Down List	134
Pull-Down Item(Necessary inCustom Only).....	135
List Box Item	135
Graffiti®	136
Non-Interactive	136
APPENDIX C: STEPS PLACED AUTOMATICALLY BY COGTOOL.....	137
APPENDIX D: SPECIAL KEYS AND SYMBOLS	139
APPENDIX E: REPORTING A BUG	140
APPENDIX F: COGTOOL SHORTCUT KEYS.....	141
APPENDIX G: COGTOOL CONTEXT MENUS	143
Project Window.....	143
Design Window	144
Frame Window	145
Script Editor.....	145
APPENDIX H: GLOSSARY.....	146
ACT-R.....	146
Demonstration.....	146
Design	146
Frame	146
Keystroke-Level Model (KLM).....	146
Project.....	147
Script	147
Task	147
Task Group	147
Transition.....	147
Widgets	147

CONVENTIONS FOR THIS USER GUIDE

CogTool runs on both Macintosh (10.3 and above) and Windows (XP and Vista) operating systems. Unless otherwise stated, this User Guide uses pictures from the Macintosh version, but it operates and looks almost the same on Windows platforms.

There are several ways to accomplish most tasks in CogTool, often using menus from the menu bar, keyboard shortcuts, or context menus. In the body of this user guide, we generally present the procedures for these tasks using the menus from the menu bar, but we also list available shortcut keys in the left margin next to the appropriate commands as well as in **Appendix G: CogTool Shortcut Keys**. We list context menus in **Appendix H: CogTool Context Menus**.

We capitalize the first letter of words that have specific meaning in CogTool, (e.g., Project, Frame, and Widget) and bold commands (e.g., From the **File** menu, choose **New Project**). When we reference another section or appendix, we write the full name (with number and subtitle) and bold it (as in the previous paragraph).

We use special graphics to denote the following items:



Notes that explain the behavior of CogTool



Tips for efficient or alternative ways to complete tasks



Warnings of potential problems and ways to avoid them



Information to remember while using CogTool



Documented bugs in CogTool (bugs are fixed more often than the User Guide is revised, and so you may not encounter these bugs in your version of CogTool)

CHAPTER 1: OVERVIEW

1.1 What is CogTool?

CogTool is a user interface (UI) prototyping tool that can produce quantitative predictions of how users will behave when the prototype is ultimately implemented. Thus, CogTool provides you with a rapid and inexpensive way to explore a large variety of UI ideas, compare them, and narrow down the options to a handful of designs to be empirically tested with users. You can rapidly analyze competitor's products as part of a competitive analysis and compare new ideas with an existing version of the system to ensure that the new design is better than the old one.

CogTool's prototyping tool follows the HCI maxim "Make frequent tasks easy and infrequent tasks possible". That is, it is fast and easy to construct a UI prototype with standard UI widgets like menus, context menus, buttons, links, check boxes, radio buttons, pull-down lists, etc. It is possible, although slightly more labor-intensive, to mock-up more unusual interfaces like pie menus or the slide bar on the iPhone. CogTool also exports prototypes to HTML, so you can share your designs with colleagues or perform quick & dirty user tests.

CogTool's quantitative predictions are based on extensive prior research in cognitive psychology. CogTool uses a "cognitive architecture" called ACT-R (Anderson and Lebiere, 1998) to simulate the cognitive, perceptual and motor behavior of humans interacting with the prototype to accomplish tasks the UI designer has defined. CogTool reliably predicts the task execution time for skilled users of the UI (John, et. al., 2004; Luo & John, B., 2005). Recent research has demonstrated the ability to predict the exploration behavior of novice users (Teo & John, 2008a,b; Teo, John & Pirolli, 2007) and the power consumption profile of mobile devices dependent on the methods user choose to accomplish tasks (Luo, 2008; Luo & Siewiorek, 2007); these research results will migrate into the released version of CogTool as time and funding allow.

1.2 How CogTool Works

There are three steps to producing quantitative predictions of human performance with CogTool:

1. Create any number of prototypes of interface designs you'd like to compare.
2. Demonstrate any number of tasks on the prototyped designs.
3. Compute human performance predictions and analyze the results.

As a prototyping tool, CogTool was inspired by James Landay's SYLK and DENIM systems (Landay and Meyers, 1995; Lin, et. al., 2000). It represents the states of a UI as a storyboard of frames; transitions between those frames represent the actions a user can take to move from one state to another. Each frame can contain widgets (e.g.,

buttons and menus) that users can act on, as well as ubiquitous input devices like a keyboard or microphone, through which a user could also change the state of the UI. A storyboard can be made from screenshots of an existing product for benchmarking or competitive analysis, from sketches of ideas you have drawn on paper or a whiteboard and then digitized, or be created directly in CogTool on a blank canvas with its widget creation tools. You can use a storyboard as documentation of your design ideas in either CogTool files (.cgt) or you can export to HTML.

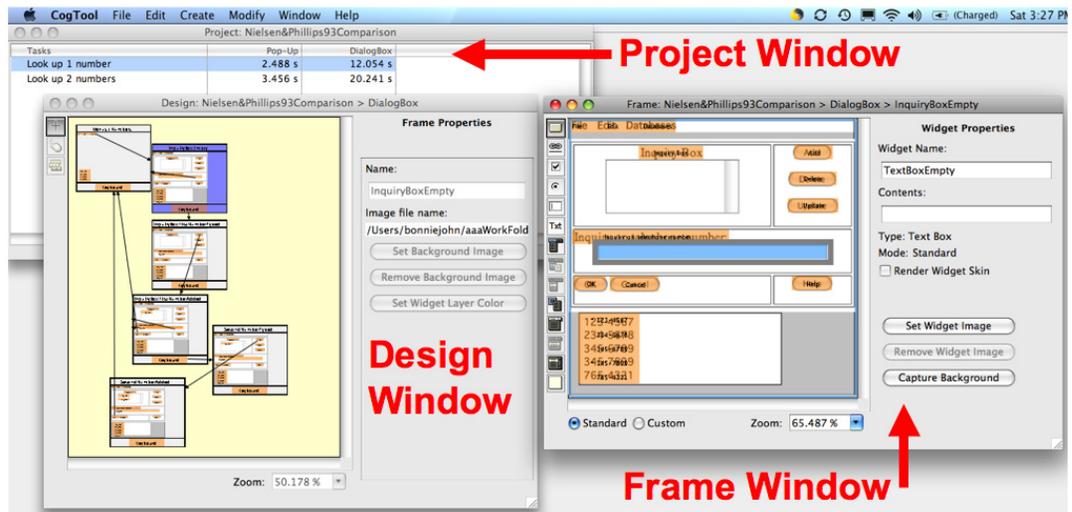


Figure 1-1: Three windows from a CogTool Project comparing two tasks on two designs. The Project window displays a grid where each column is a different design, each row is a different task, and the intersection of each column and row shows the prediction of skilled execution time for that task on that design. The Design window is where frames and the transitions between them are created. The Frame window is where widgets are placed to represent how the design appears in a particular state. (Designs and tasks in this example are from Nielsen & Phillips, 1993)

To make predictions of how long it will take a skilled user to execute a task on a UI design, simply demonstrate the task on the design. CogTool uses the storyboard and the demonstration to produce a computational cognitive model of a skilled user's performance. The model that CogTool creates is based on the Keystroke-Level Model (KLM; Card, Moran, and Newell, 1980) and implemented "under the hood" in ACT-R (Anderson and Lebiere, 1998). KLM has been verified to be reliable within 20% of observed human performance times in over 100 papers in the HCI literature. CogTool inherits at least this accuracy, but is likely to be even more accurate because it generates the model automatically, removing the need for the analyst to be trained in the theory and practice of KLM (John, et. al., 2004).

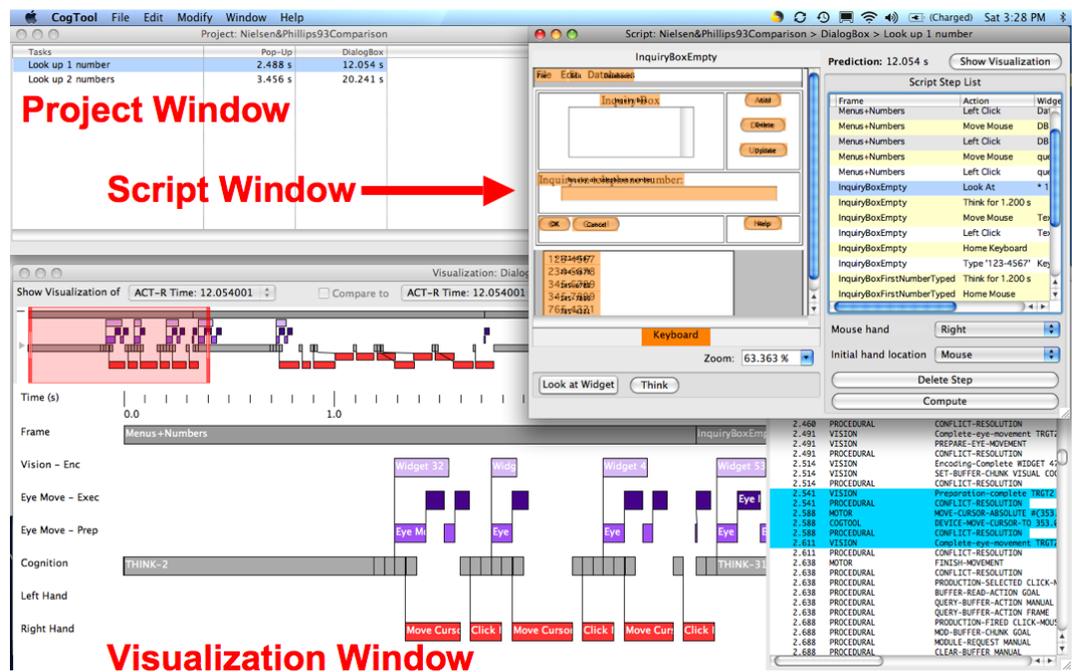


Figure 1-2: Three other windows from a CogTool Project comparing two tasks on two designs. The Project window, as above, displays the grid of designs, tasks and predictions. The Script window is where CogTool records your demonstration of a task on a design and creates the cognitive model. The Visualization window is you can see what ACT-R is doing “under the hood” to produce the prediction.

1.3 Uses of CogTool

As a prototyping tool, CogTool design storyboards can be used to

- Build and record design ideas in an interactive form
- Communicate interactive behavior to development teams or management
- Produce HTML for quick-and-dirty user testing.

While CogTool predictions are not intended to replace all user testing if accurate estimates of skilled performance time is highly valuable to your enterprise (e.g., if every second of a users time translates into hundreds of thousands of dollars a year of revenue or expense, or if a split second difference in performance time can save a life), they can be used to

- Compare design ideas, allowing you to explore widely, but only conduct expensive user tests with the few most promising ideas.
- Compare competitors’ products to your own during a competitive analysis or for marketing purposes.
- Provide quantitative benchmarks for internal use during development to ensure that a new design is at least as efficient as the previous version.
- Determine requirements during acquisition and test that the delivered system will

meet those requirements.

- Analyze an existing system for bottlenecks in user behavior, focussing your redesign effort on the aspects of the interface that need it most.

We hope you find CogTool useful in your work and would love to hear from you with testimonials or suggestions. If you would like to share your experience with us, please visit the CogTool website (<http://cogtool.org/>) and participate in our User Forums or send email to cogtool@cs.cmu.edu.

CHAPTER 2: GETTING STARTED WITH COGTOOL

2.1 Downloading CogTool

The latest public release of CogTool is available from the CogTool Project's website on the download page (<http://cogtool.hcii.cs.cmu.edu/use-today/download-cogtool>).

CogTool is available for Windows XP/Vista and Mac OS X versions 10.4.8 and later.

2.2 Installing CogTool

2.2.1 Installing on Mac OS X

- When you download the CogTool file from the website, your browser will launch archive software, such as Stuffit, to unpack the CogTool executable. If it does not, double-click the downloaded CogTool file to expand it.



The CogTool file is expanded in the same location as the downloaded file, which in most cases is the Desktop.

- Place the expanded CogTool application file wherever you like. Although many people prefer to put it in the **Applications** folder, CogTool will run from any location on your hard drive or writable external storage media.

2.2.2 Installing on Windows XP and Vista

- You must have the Java Runtime Environment (JRE 1.5 or later) installed in order to use CogTool. If you do not know if you have JRE installed, you can test it by opening this link in your web browser: <http://www.java.com/en/download/help/testvm.xml>. This page will automatically test your computer and tell you whether JRE is installed or not. If you do not already have JRE installed, the latest release and instructions on how to install it are available at <http://www.java.com/en/download/>.
- Double-click the CogTool installer to launch the install wizard.



- Follow the instructions in the wizard to complete the CogTool installation. You can install CogTool where ever you wish on you hard drive.
- Once CogTool is installed, it can be launched from the Windows Start Menu, or double-clicked where ever its icon appears.

2.3 Launching CogTool

You can launch CogTool by double-clicking the application icon or by choosing the application from the listing in the Windows Start Menu.

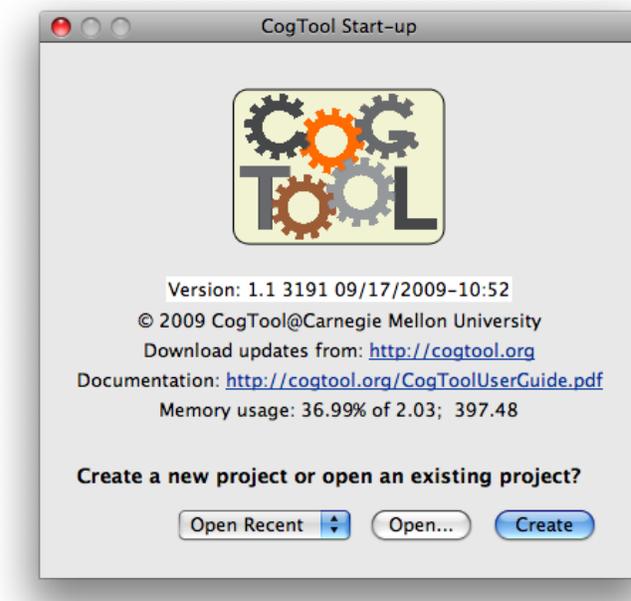


Figure 2-1: The first window you see when you launch CogTool.



Double-clicking on a saved project file (.cgt file) to open it is not recommended. In Mac OS X, it does nothing, in Windows it opens a new instance of CogTool even if one is already running. Instead, first launch CogTool. then click the **Open** button and navigate to the saved CogTool file. If you have been working with the project recently, it will show up in the **Open Recent** pull-down list.



With Mac OS X, you can launch the CogTool application from any location to which you save it.

2.4 Exploring CogTool

Although CogTool is not intended to be a walk-up-and-use system, especially if you want to get cognitively plausible quantitative predictions of skilled performance time, CogTool's user interface provides common facilities to aid exploration.

- *Menus.* Almost all commands are accessible through the menus at the top of the window (Windows) or screen (Mac). Menus are tailored to the currently active window, so explore them all. If there are keyboard shortcuts for the command, they are listed in the menus.
- *Context Menus.* The most common commands used in a window or on an object are available in a context menu invoked by right-clicking (Windows) or CTRL-clicking (Mac). They are listed in Appendix G.
- *Roll-overs.* You can roll over many objects in CogTool to get more information about them. For example, roll-over the tools in the Widget or Transition to see their names. Roll-over a Transition in the Design window to see its type, source and target. Roll-over a Widget anywhere to see its display label, name and type.
- *Double-clicking.* Double-clicking something tends to open it for editing. In the Project windows, double-clicking on a Design name opens a window to edit that Design; double-clicking a cell opens its Script window. In a Design window, double-clicking a Frame opens a window for editing that Frame. In a Script window, double-clicking the Frame name opens the Frame editor; double-clicking a Think operator allows you to edit its label and duration. You get the idea - if you want to edit something, try double-clicking it.

and most important,

- *Undo.* Works. Multi-level. Fear not -- you can undo anything you've done, so explore with impunity.

2.5 CogTool Help

We do not currently have on-line help for CogTool;. However, user forums are available at <http://cogtool.org/> where you can get questions answered and see what other people do.

We encourage you to have this pdf document open in Adobe Reader as you work in CogTool. Be sure to use the Bookmarks feature of Reader, so you can navigate easily around the User Guide. We have enabled commenting, so you can mark up the document as you like.

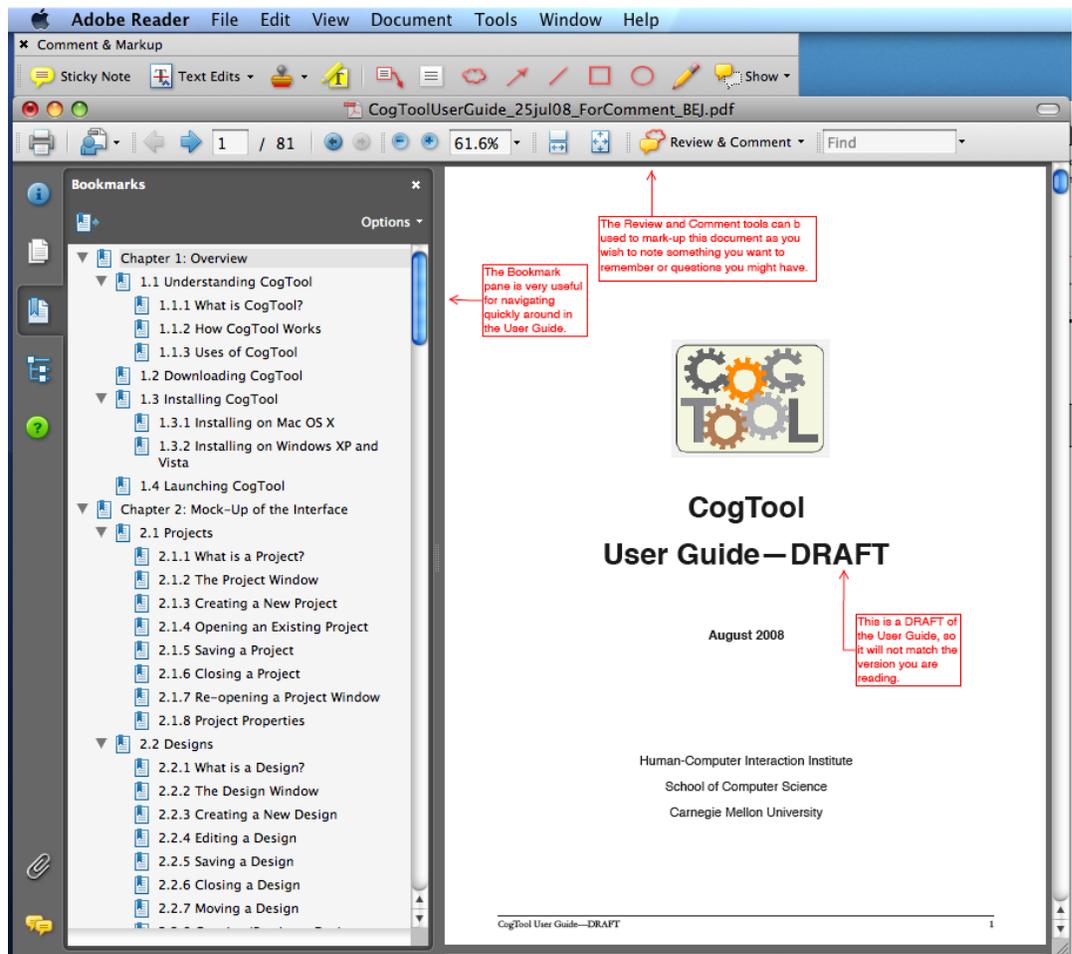


Figure 2-2: The CogTool User Guide opened in AdobeReader (v7 or later) with the Bookmarks pane and Review and Comment toolbar.

If you are running Mac OSX 10.5 or later, we encourage you to explore the automatic help for menu items. Consider the case where you are trying to align several widgets (highlighted in purple) but you have forgotten where the alignment commands are in the menus. If you open the Help menu in MacOSX and search for “align”, it lists all the menu items that contain “align” and if you roll over one of them, it opens the correct menu and points to the command (pointer in dark blue).

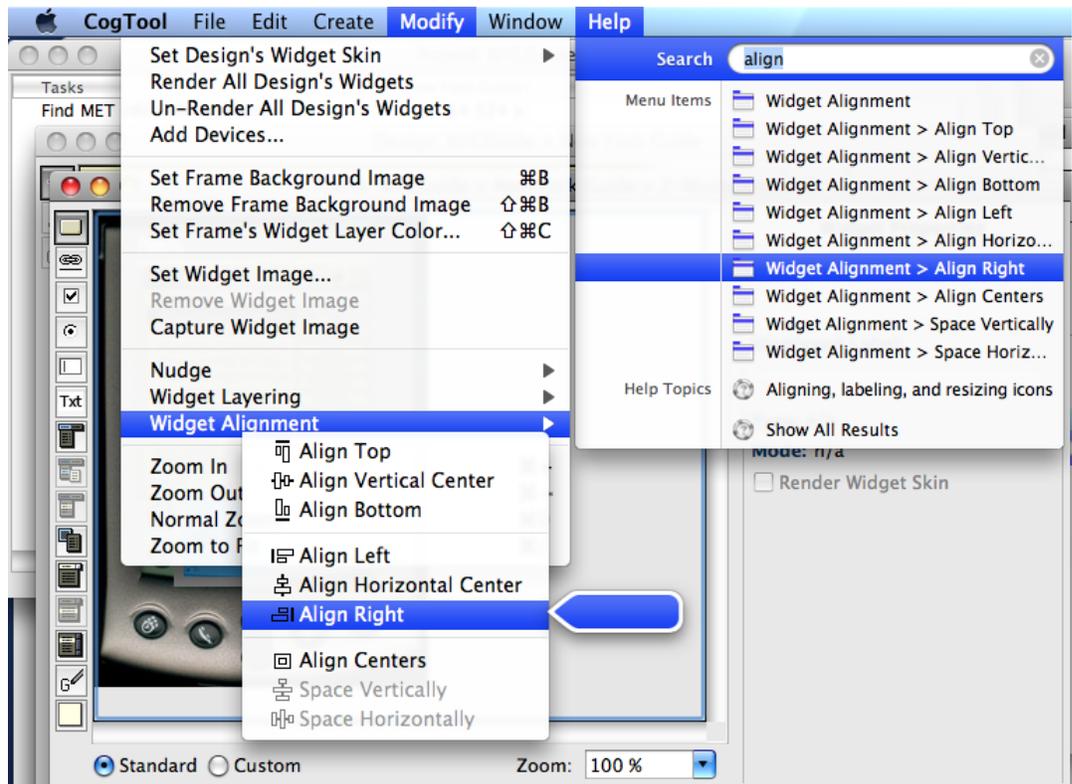


Figure 2-3: Help finding menu commands is automatically provided by Apple's Mac OS X 10.5 or later. Menu change depending on which type of window is active, so explore different windows when looking for a particular command.

CHAPTER 3: MANAGING A PROJECT

3.1 What is a Project?

A CogTool Project stores all the interface designs, tasks that can be performed on the designs, and predictions of user behavior. Projects are stored in files called cgt (CoG-Tool) files when a Project is saved.

 By default, the Project name is also the name of the .cgt file created by CogTool. However, when the save box appears, you can save the file with another name.

3.2 The Project Window

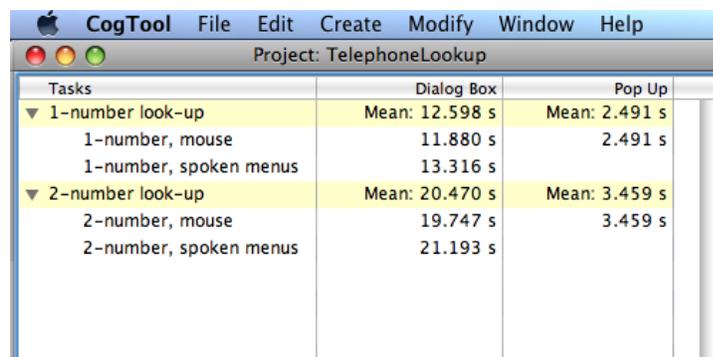
The Project window is laid out in a grid where the columns are Designs and the rows are Tasks that can be performed on those Designs (See Figure 3-1).

 The title bar of a Project window starts with the name “Project” and contains the name of the Project.

Each column header, starting with the second column, is the name of an interface design you have prototyped in the Project.

The row headers are the names of the different Tasks and Task Groups.

The intersection of each Design and Task will show the results of a quantitative prediction you have made using CogTool (See Figure 3-1).



Tasks	Dialog Box	Pop Up
▼ 1-number look-up	Mean: 12.598 s	Mean: 2.491 s
1-number, mouse	11.880 s	2.491 s
1-number, spoken menus	13.316 s	
▼ 2-number look-up	Mean: 20.470 s	Mean: 3.459 s
2-number, mouse	19.747 s	3.459 s
2-number, spoken menus	21.193 s	

Figure 3-1: The Project window: This example has two Designs and two Task groups. The Dialog Box design allows menus items to be selected either by mouse or by speech input so predictions for these two methods are shown for both Tasks and they are grouped. The Pop-Up design only uses the mouse, so only one prediction is shown for each Task Group.

3.3 Creating a New Project

To create a Project, do one of the following:

- In the CogTool startup window, click the **Create** button.

OR

- From the **File** menu, choose **New Project**.

Mac: ⌘N

Win: CTRL+N

Because a Project must contain at least one Design, you will be prompted to name a new Design for the new Project (see Figure 3-2). The name you choose should be descriptive of the interface being modeled. Since each Design must have at least one input and one output device, you will be prompted to select devices in the **New Design** dialog box. See **Section 4.1.3 Creating a New Design** for more information on the **New Design** dialog box.

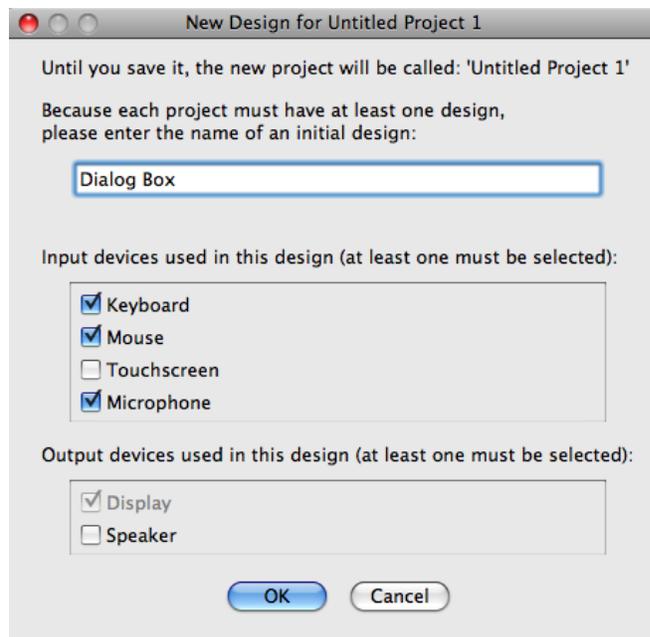


Figure 3-2: The New Design dialog box: This example is creating the Dialog Box Design shown in the previous figure, entitled Dialog Box, with a keyboard, mouse and microphone as input devices and a display as an output device.



CogTool_v1.0beta23 assumes every Design has a display. Thus, Display is selected as an output device and it is not possible to deselect the display. (If your Design has no display, e.g., is audio only, its Frames can remain empty to simulate having no display.)

Hitting the OK button results in a new Project window, with the Design you just named and its Design window open ready to edit the Design. At this point you can start creating the contents of your Design (editing a Design is discussed in a subsequent chapter.)

or return to the Project window to set up additional Designs.

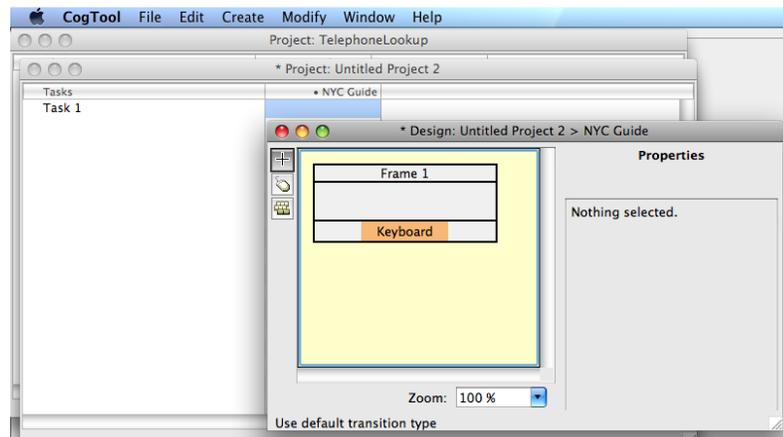


Figure 3-3: These windows appeared after creating a new Project with a Design called “NYC Guide” that has a keyboard and mouse as input devices.

3.4 Opening an Existing Project

To open an existing Project, do one of the following:

- In the CogTool startup window, click the **Open** button.

OR

- If you have been editing the Project recently, in the CogTool startup window, click the **Open Recent** pull-down list and select the desired Project from the list.

OR

- From the **File** menu, choose **Open Project**.

Mac: ⌘O

Win: CTRL+O

You will be prompted to choose a file to open.

In addition, CogTool keeps a list of the last 10 Projects you have opened in its list of recently opened Projects accessible after the start-up screen.

To reopen recent Projects after the start-up screen:

- From the **File** menu, choose **Open Recent**.
- Select the file name from the list.

3.5 Saving a Project

To save a Project:

- From the **File** menu, choose **Save Project**.

Mac: ⌘S

Win: CTRL+S

The first time you save a file, you will be prompted for the filename and location. The file will be given a .cgt extension. The filename (minus the .cgt extension) will appear in the title bar of all CogTool windows for the Project.

 An asterisk (*) in front of the word “Project” in the title bar of a window signifies that the Project has unsaved modifications (see Figure 3-4). When the file has been saved, the asterisk disappears.

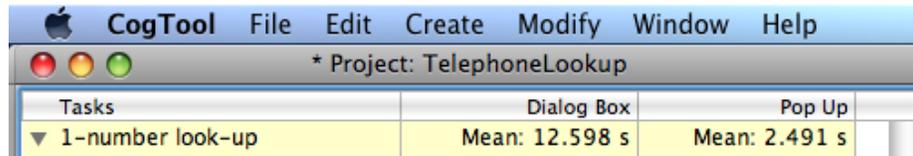


Figure 3-4: The asterisk in the tile bar denotes a Project with unsaved modifications.

To save a Project with a new name:

Mac: ⌘S

Win:

CTRL+SHIFT+S

- From the **File** menu, choose **Save As**.
- You will be prompted for a filename and location

3.6 Closing a Project

To close a Project:

- From the **File** menu, choose **Close Project**.

To close a window:

- From the **File** menu, choose **Close Window**.

Mac: ⌘W

Win: CTRL+W

 If a Project has unsaved modifications, you will be prompted to save the Project when you close it.

 If you close the last window belonging to a Project, closing the window also closes the Project. If that Project has unsaved modifications when you close the last window, you will be prompted to save the Project.



You can also close an open window by clicking the close button on the window title bar.

3.7 Project Properties

To see the version number of the CogTool application last used to save the Project and the Project name:

- From the **File** menu, choose **Properties** (see Figure 3-5).



Figure 3-5: A sample Project Properties window

3.8 Reopening Project Windows and Understanding the Windowing System

As you create a design prototype, you will work in the Project window, a Design window and many Frame windows. As you make predictions of human performance on your designs, you will work in the Project window, a Script window, and the Visualization window. Each of these windows will be described in their own sections of this User Guide, but understanding their relationship may help you navigate the many windows you create as you work.

The Project window is considered the top of the hierarchy of windows and it owns all the other windows associated with that Project. A Design window owns all the Frame windows that depict the states of the display in that design. Script windows and Visualization windows are owned by their Project.

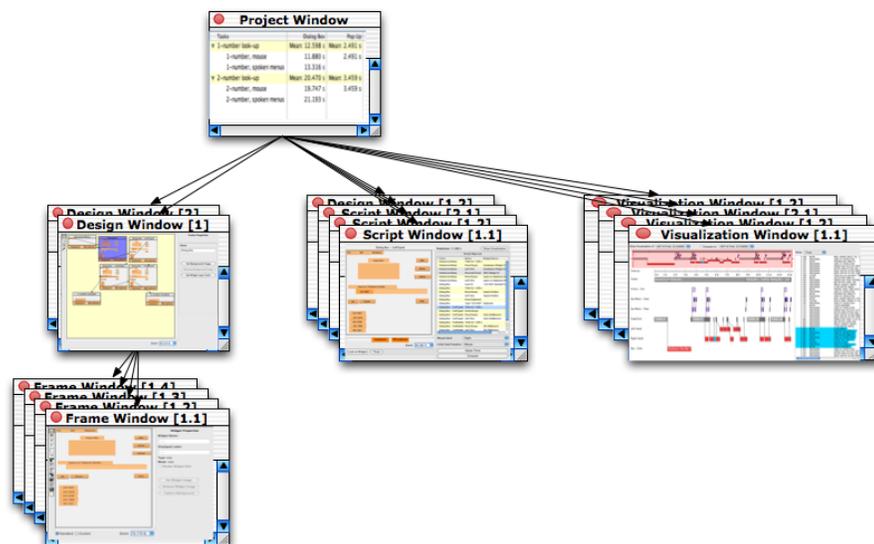


Figure 3-6: The relationship of windows in a Project that has two Designs and four tasks demonstrated in Script windows. Each Design may include many Frame windows. The result of computing each script can be shown in a Visualization window.

The **Window** menu reflects this relationship, and you can get to any open window, or

any Project window even if it has been closed, through the **Window** menu. The top part of the Windows menu lists the windows above the active window in the hierarchy. In this case, a Frame window is active, so its Design and Project windows are listed in the top part of the Windows menu. The bottom part of the Windows menu lists all the open Projects and this expands to all the open windows in those Projects. In the Figure, only one project is open, but many windows in that Project appear.

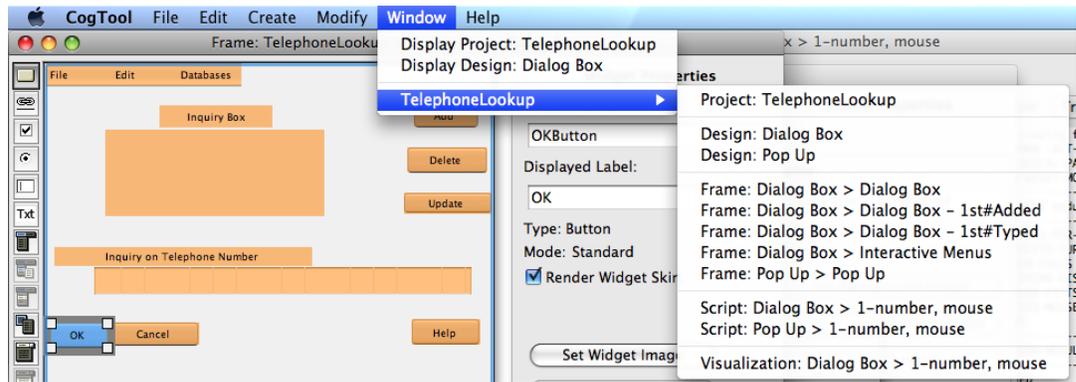


Figure 3-7: View of the Window menu

Therefore, if you closed a Project window and want to reopen it, make any of its Design, Frame, Script, or Visualization windows is active, then:

- From the **Window** menu, choose **Display Project**.

To bring any open window to the top and make it active:

- From the **Window** menu, choose the desired Project name and it will expand to list all its open windows, labeled as to which type they are.
- Choose the desired window from the list.

CHAPTER 4: PROTOTYPING AN INTERFACE

The prototyping aspect of CogTool was inspired by James Landay's SYLK and DEN-IM systems (Landay and Meyers, 2005; Lin, et. al., 2000). It represents the states of a UI as a storyboard of frames; transitions between those frames represent the actions a user can take to move from one state to another. Each frame can contain widgets (e.g., buttons and menus) that users can act on, as well as ubiquitous input devices like a keyboard or microphone, through which a user could also change the state of the UI.

A storyboard can be made from screenshots of an existing product for benchmarking or competitive analysis, from sketches of ideas you have drawn on paper or a whiteboard and then digitized, be created directly in CogTool on a blank canvas with its widget creation tools, or even imported from a HTML on your computer or the Web. You can use a storyboard as documentation of your design ideas in either CogTool files (.cgt) or you can export to HTML.

This chapter will present how to build a design so that it not only records your design ideas, but can be used to make predictions of human performance (discussed in the next chapter).

4.1 Designs

4.1.1 What is a Design?

A Design is a prototype of the system you are proposing to build. It represented the system as a series of Frames and the user actions that transition between them. Each Frame represents how the interface looks to a user. You can represent just the display of a computer monitor in a frame and associate input Devices with that display (e.g, a keyboard and a mouse, a speaker and microphone), or you can represent an entire device in a Frame (e.g, the display and hard buttons of a cell phone). Transitions, which represents user actions, cause a Frame to change to another Frame. Together, the Devices, Frames, and Transitions make up a Design.

You work on Designs in the Project window and each Design's own window. You work with a Design's definition in the Project window (e.g., creating, renaming, moving, deleting), and with the Design's content in each Design's own window.

4.1.2 What is a Device?

A Device is a representation of the hardware associated with you Design. CogTool can represent a fixed set of Devices. Input Devices include keyboard, mouse, touchscreen and microphone. Output Devices include a display and a speaker. The Devices associ-

ated with a Design influence the Widgets that can be include in Frames and the Transitions that can change one Frame to the next. For example, if there is no microphone included in a Design, there can be no Transitions based on spoken commands. You choose Devices for a Design when you create that Design. If you want to add more Devices later, you can. However, you cannot remove Devices from a Design.

See Appendix A for a list of the available Devices, definitions, and tips of how to use these devices to prototype a wide range of interactive systems.

4.1.3 The Design Window

The Design window shows the Frames and the Transitions between them. Design windows have a light yellow background to distinguish them from the other CogTool windows. The title bar starts with the word “Design” (see Figure 4-1).

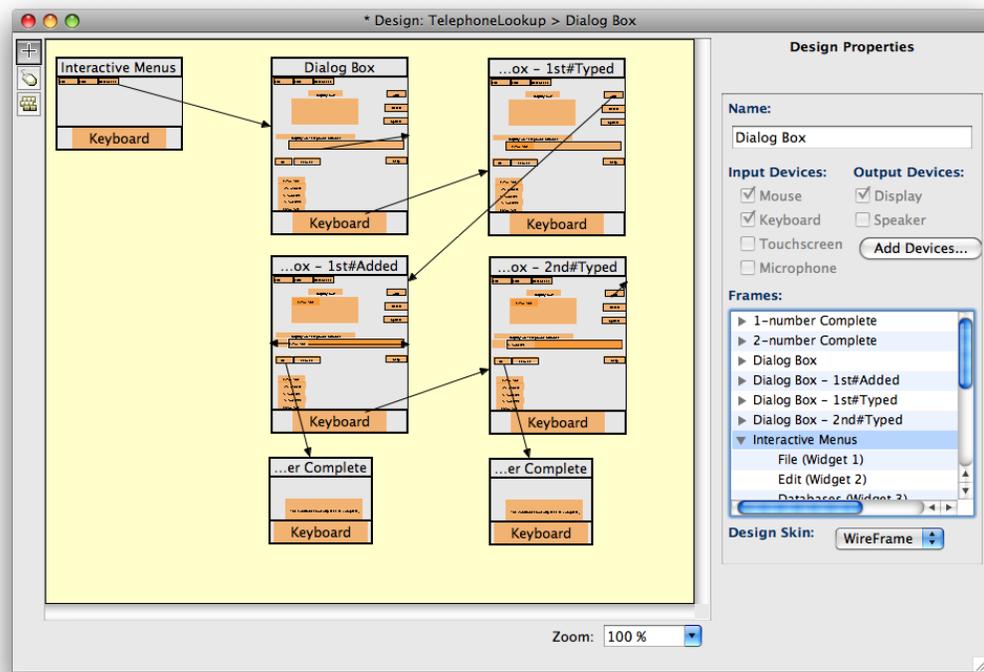


Figure 4-1: The Design Window: When no Frame is selected (no Frames are blue), the properties of the Design appear in the pane on the right.

The Design canvas is the main area of the Design window, where Frames are added and connected by Transitions. When no Frame or Transition is selected, as in the figure above, the properties of the Design appear in the properties pane on the right. Its name, input devices, output devices, a list of its Frames that can be expanded to show the Frame’s Widgets and Transitions, and the type of skin that will be applied to rendered Widgets. When a Frame or Transition is selected in the Design canvas, its properties appear in the properties pane, discussed in the Frames and Transitions sections. The toolbar on the left holds tools for drawing Transitions, discussed in the Transition

section.

4.1.4 Creating a New Design

Each Project must have at least one Design, which is created when a new Project is created. However, because CogTool allows you to compare human performance across different designs, you are likely to want more than one Design for your Project.

To create additional Designs for your existing Project:

Mac: ⌘D

Win:

CTRL+SHIFT+D

- In the Project windows, from the **Create** menu, choose **New Design**. The New Design dialog box will appear.

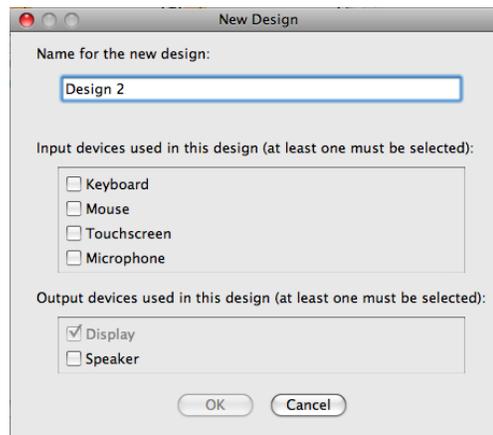


Figure 4-2: The New Design dialog box. The OK button will not be active until at least one input device is checked.

In this dialog box, do the following:

- Choose a name unique within the Project and descriptive of the interface being prototyped. This will help you to identify the purpose of the Design when you have multiple Designs in your Project.



Text in CogTool's item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

- Check the input and output devices that your Design should include. You can check as many devices as you need to represent your Design. Typical computer systems will have a keyboard and mouse as input and a display as output. A PDA may have a touchscreen. Speech recognition systems will have a microphone, etc. You will not be able to click **OK** until you have selected at least one input device.



CogTool assumes every Design has a display. Thus, Display is selected as an output device and it is not possible to deselect the display. If your device has no display (e.g., is audio only), this can be prototyped by all Frames being empty.



If you are prototyping a cell phone or other hand-held device that is operated with the fingers, use “Touchscreen” as the input device, even if the buttons are physical buttons not on an actual touchscreen. This will allow the user’s actions to be represented as tapping, which is sufficiently close to pressing a physical button to make human performance predictions.

When Designs are created, the Design columns appear in the Project window in the order in which they are created, with new columns appearing at the far right. If a Design is selected when a new Design is created, the new Design will appear immediately to the right of the selected Design column (see Figure 4-3).

Tasks	Dialog Box	• Design 3	Pop Up
1-number lookup	11.880 s		2.441 s
2-number lookup	19.747 s		3.409 s

Figure 4-3: The Dialog Box Design was selected when Design 3 was created.

You can also create a design by importing it from HTML, either on your computer or from the Web. This is useful if your design process already includes interactive prototypes expressed in HTML or if you want to analyze existing web sites. This feature is discussed in Section 4.6.

4.1.5 Selecting a Design

In the Project window, click on a Design name to select the Design. A dot will appear to the left of the Design name and the column will turn blue to indicate selection. You cannot select multiple Designs.

4.1.6 Editing a Design

To open a Design window so you can edit a Design, do one of the following:

- In the Project Window, select the Design.
- From the **Edit** menu, choose **Edit Design**.

Mac: ⌘E

Win: CTRL+E



A Design is selected when a dot appears next to the Design name and the Script cells below the name are highlighted in blue (see Design 3 in the previous figure).

OR

- Double-click the Design name in its column in the Project window.

A Design window appears where you can add Frames and Transitions to prototype your system (see Figure 4-4). Editing Frames and defining Transitions are discussed in

subsequent sections.

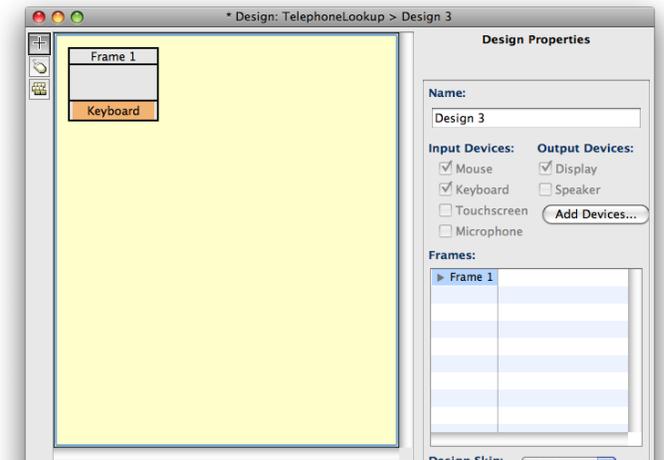


Figure 4-4: The Design window that results from opening a new Design. The Design canvas has one empty Frame and nothing is selected.

4.1.7 Saving a Design

Changes to your Design are saved when you save the Project. Choose **Save Project** from the **File** menu, or use the shortcut key, at any time to save your work.

4.1.8 Closing a Design

To close a Design window, when it is active:

- From the **File** menu, choose **Close Window** (or hit the close window button provided by the operating system on every window).

Mac: ⌘W

Win: CTRL+W

4.1.9 Moving a Design

You can change the order of the Design columns in the Project window by clicking on the Design name and dragging the column to the desired position. The Design you are dragging become transparent, which allows you to see when you have reached the desired position.

The image shows a window titled "CogTool" with a menu bar (File, Edit, Create, Modify, Window, Help) and a title bar "* Project: TelephoneLookup". Below the title bar is a table with columns "Tasks", "Dialog Box", and "Pop Up".

Tasks	Dialog Box	Pop Up
1-number lookup	1.880 s	2.441 s
2-number lookup	19.747 s	3.409 s

Figure 4-5: The Pop Up Design is being moved from the right-most position to the left of the Dialog Box Design. You can see the Dialog Box column underneath as you

move the Pop Up Design.

You can also change the order of the Design columns by cutting and pasting the Design columns into the desired location.

4.1.10 Cutting, Copying, and Pasting a Design

You can cut or copy a Design and paste it into another location of the same Project or into a different Project.

To cut a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Cut Design**.
The Design is now on the clipboard, ready to be pasted.

Mac: ⌘X

Win: CTRL+X

To copy a Design:

- In the Project window, select the Design.
- From the **Edit** menu, choose **Copy Design**.
The Design is now on the clipboard, ready to be pasted.

Mac: ⌘C

Win: CTRL+C

To paste a Design:

- In the Project window, from the **Edit** menu, choose **Paste**.



If no Design is selected, the pasted Design will be added to the right-most column. If an existing Design is selected, the pasted Design will be added immediately to the right of the selected Design.

If the Design's name is already in use in the Project, a unique suffix is appended to the pasted Design's name (see Figure 4-6).

Tasks	Dialog Box	Pop Up	• Pop Up [2]
1-number lookup	11.880 s	2.441 s	2.441 s
2-number lookup	19.747 s	3.409 s	3.409 s

Figure 4-6: CogTool added a unique suffix to the name of the Pop Up Design in the right-most column when it was pasted into the same Project.

When pasting into a different Project, if the Design already has Tasks defined in it and one or more of the Task names already exist in the target Project, they will be pasted into the existing Task. If this is not what you intended (e.g., Tasks named the same are actually different), create different Task names and move the pasted Tasks to their proper locations. If the target Project does not have exactly the same Task names, pasting the Design will create new Task names in the target Project. Again, if this is not what you intended (e.g., differently named Tasks are actually the same), move the pasted cells of the Design to the desired Task rows. See Tasks, later in this document for more

information on creating and moving Tasks.

4.1.11 Deleting a Design

To delete a Design:

Mac: ⌘⌫
Win:
CTRL+DELETE

- In the Project window, select the Design.
- From the **Edit** menu, choose **Delete Design**.

You will have to confirm the deletion of the Design. It will not be deleted until you click **OK** in the confirmation dialog box.



Deleting a Design has the same effect as cutting a Design, except that the Design is not on the clipboard and cannot then be pasted.

4.1.12 Duplicating a Design

To duplicate a Design:

Mac: ⌘D
Win: CTRL+D

- In the Project window, select the Design.
- From the **Edit** menu, choose **Duplicate Design**.



The Design is duplicated to the right of the original Design, and its name will be given a unique suffix (e.g. “MyDesign[1]”).



Within a Project, duplicating a Design has the same effect as copying and pasting a Design. However, duplicating a Design does not put it on the clipboard and you cannot then paste it into a different Project.

4.1.13 Renaming a Design

To rename a Design:

Mac: ⌘R
Win: CTRL+R

- In the Project window, select the Design.
- From the **Edit** menu, choose **Rename Design**.

Enter the new name for the Design when prompted. The name will be changed when you click **OK** (see Figure 4-7).

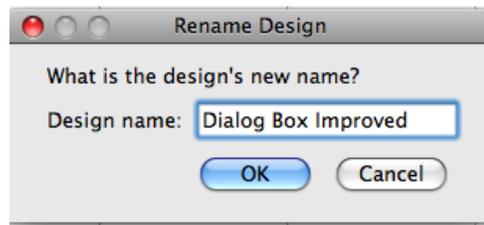


Figure 4-7: The Rename Design dialog box



Each Design name must be unique within a Project.



Text in CogTool's item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

4.1.14 Bringing a Design Window to the Foreground

To bring a Design window to the foreground, do one of the following:

- In the Project window, double-click the Design name.

OR

- From the **Window** menu, if the Design name is available as the second item in the menu, choose it.

OR

- From the **Window** menu, if the Design name does not appear directly, choose the Project name below the separator. A submenu will appear.
- From the submenu, choose the Design name.

Zoom In

Mac: ⌘=

Win: CTRL+=

Zoom Out

Mac: ⌘-

Win: CTRL+-

Normal Zoom

Mac: ⌘0

Win: CTRL+0

Zoom to Fit

Mac: ⌘/

Win: CTRL+/

4.1.15 Zooming a Design

When in a Design window, you may zoom in and out on the Design to make working with it easier. This is often necessary when your Design has many Frames.

To zoom a Design, do one of the following:

- From the Modify menu, choose the Zoom option you want.

OR

- Choose your zoom settings from the box at the bottom of the Design window.



Normal Zoom makes the contents of the Frame revert to the default size. **Zoom to Fit** makes the contents of a Frame fit the available space in the window.

4.1.16 Adding Devices to a Design

To add a device to a Design:

- In the Project window, select the Design.
- From the **Modify** menu, choose **Add Devices....**
The Add Devices dialog box will appear.

OR

- Click the Add Devices... button in the properties pane of the Design window.
The Add Devices dialog box will appear.
- Check the devices you want to add and click **OK**.



Figure 4-8: The Add Devices dialog box with one new device to be added.



You cannot undo the addition of a device, so choose carefully. However, the only consequence of having more devices than you need is irrelevant items in the Widget toolbar and on the Frames. It does not prevent you from working, but it might be more difficult to explain the irrelevant information to colleagues.

4.2 Frames

4.2.1 What is a Frame?

A Design consists of Frames and the Transitions that link them. A Frame represents a single screen of a user interface. A Transition represents a user's action that changes one Frame into another. Frames may contain background images and Widgets to represent the look and interactivity of the interface you are designing.

You work on Frames in both the Design window and each Frame's own window. You

work with a Frame's definition in the Design window (e.g., creating, renaming, deleting), and with the Frame's content in each Frame's own window.

4.2.2 The Frame Window

The main components of the Frame window are a toolbar of Widgets on the far left, a Frame canvas that contains the interface representation in the center, and a properties pane on the right.

The Widget types listed on the toolbar to the left of the window are available depending on (1) the devices associated with the Design and (2) whether the Standard or Custom toolbar option is selected (see Figure 4-9). For example, if you have associated a touchscreen with a Design, then the Graffiti widget appears in the toolbar, but is not in the toolbar if your Design only has a mouse and keyboard. If you have selected the Standard widgets, entire interactive menu structures are constructed for you automatically, so you don't need the widgets for custom menu items and submenus and they are grayed out. (See subsequent sections for more information on Widgets and the difference between Standard and Custom).

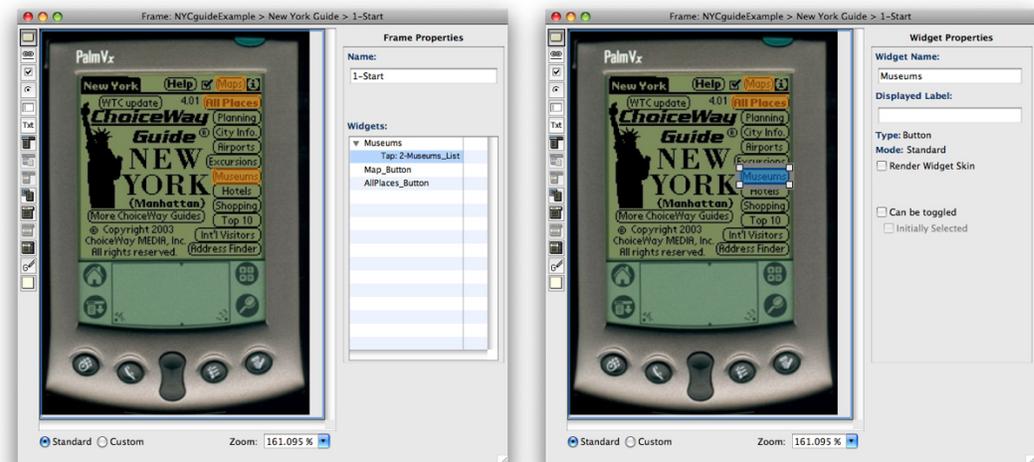


Figure 4-9: Two different views of the Frame window with a Design for a NYC Guide on a PalmVx. This Frame has a background image (the PalmVx) and three standard button widgets have been defined (colored orange, a settable color). On the left, no widgets are selected, so the Properties pane on the right displays the Frame Properties; its name and a list of the widgets and transitions already defined in the Frame. On the right, the Museums is selected, is highlighted in a blue and the Properties pane displays the Widget's properties instead of the Frame's properties. Both views show the Standard radio button selected (bottom left), so the toolbar has some grayed out tools unnecessary for standard widgets. (See subsequent sections for more information on the difference between Standard and Custom widgets.)

The properties pane on the right contains information about the Frame itself if no Widgets are selected or the Widget that is selected in the Frame canvas. If more than

one Widget is selected, the Widget properties pane presents “n/a” (not applicable) in its fields.

The Frame properties include its name in an editable field and a list of all the Widgets defined in that frame (see Figure 4-9). Click on the expansion arrow to show to the left of a Widget name to see the Transitions associated with that Widget. Clicking on the Widget name will select it, scroll the canvas to bring that Widget into view, highlight that Widget in the Frame, and displaying its properties.

The Widget properties include its name, whatever text is displayed as its label (if any), and properties specific to each Widget type. (See subsequent sections for more information about Widgets and their properties.)



When a Frame window is made too narrow, the Standard, Custom, and Zoom labels run into each other (see Figure 4-10). When it is too short, buttons on the bottom of the Properties pane disappear.

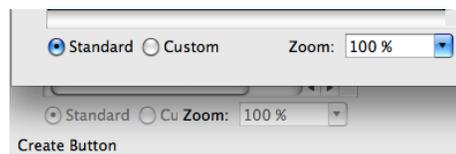


Figure 4-10: Example of a Frame that is sufficiently wide to see all the buttons (top) and a Frame that is so narrow that the labels run together (bottom).

4.2.3 Setting the Default size of a Frame

The default size of a frame is a good aspect ratio for prototyping systems that run on a desktop or laptop. However, it may be too wide for some devices, like clam-shell cell-phones which are far taller than they are wide when open. You can set the minimum width of new frames in the Preferences dialog box (under the CogTool menu on a Mac; under File in Windows). Once set, each new Frame will be created at this minimum width. This is the minimum width - you can always create wider Frames by placing wider background images in them or placing Widgets further to the right.



The Preferences dialog box contains a button for accessing research commands. These are undocumented, untested, and unvalidated. Explore at your own risk .

4.2.4 Creating a New Frame

When a Design window is opened for the first time, a new Frame is automatically created.

Mac: ⌘F

Win:

CTRL+SHIFT+F

To create a new Frame at any time:

- In the Design window, from the **Create** menu, choose **New Frame**. A new Frame appears in the Design window. It is selected, has been given a unique name by default, and this name is highlighted and ready for you to type in a meaningful name.



Text in CogTool's item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

4.2.4.1 Creating new Frames from a folder of images

You may have a folder of images of your interface that you want to import. This may be created if you sketched your interface screens on paper and scanned them in, or sketched it on a whiteboard and took digital pictures with a camera, or created wireframes in a drawing program, or even took screen shots from an earlier version or competitor's product. If so, CogTool provides an easy way to import all the pictures as background images in multiple Frames.

You can import the entire folder of images for use in your Frames. Each image will become the background of its own Frame. The Frame's title will be the same as the corresponding file name, without the extension.



Only JPG, PNG, or GIF are acceptable file formats for CogTool.

To import background images:

- Open the Design window.
- From the **Modify** menu, choose **Import Background Images**.
- Navigate to the folder that contains the images.
- Click **Choose**.



If the Design contains only the default Frame, the default Frame will be deleted when the images are imported.

The Frames will be imported in alphabetical order, in rows from left to right, starting at the top right of the Design canvas. You can set the number of Frames in each row in the Preferences menu (under CogTool on the Mac; under File in Windows).



If you have a lot of images you wish to import, consider beginning their names with numbers ordering them in the way you want them to appear in your Design. If there are more than 9 and less than 100, name them 01_ImageName1, 02_ImageName2, 03_ImageName3, etc.

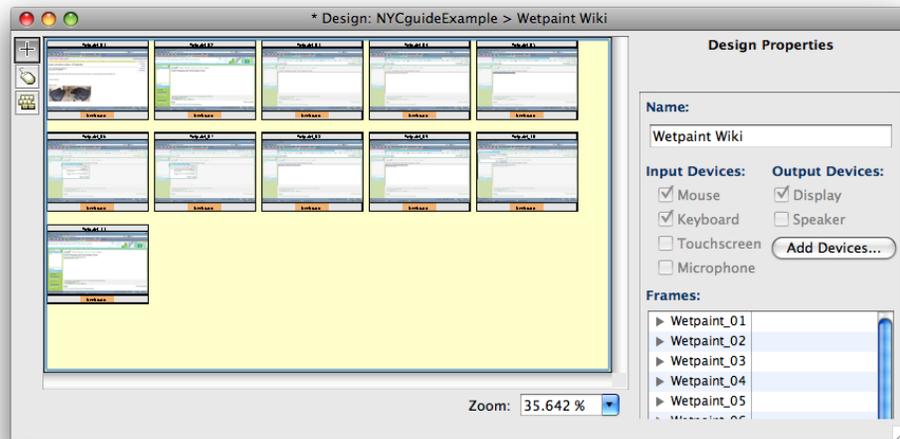


Figure 4-11: The Design window after 11 frames have been imported. The Frames are imported in alphabetical order and the preferences were set to import 5 frames in each row.



The number of Frames in each row of the Design window created by importing can be set in the Preferences dialog box (under the CogTool menu on a Mac; under File in Windows).



The Preferences dialog box contains a button for accessing research commands. These are undocumented, untested, and unvalidated. Explore at your own risk .

4.2.4.2 Creating new Frames containing a “template” of Widgets

When you are prototyping a system that has many of the same widgets in multiple Frames, e.g., a cellphone with many hard buttons or a airline cockpit device like the Control Display Unit where only the screen changes from Frame to Frame, CogTool provides an efficient way for you to create each new Frame already containing the same widgets.

To create a Frame Template of Widgets

- In any Frame window, select the Widgets you want to be on every new Frame in the Design.
- From the Modify menu, choose Set Frame Template.
- Create new Frames, and each of them will be pre-populated with the Widgets in the template.



Figure 4-12: All the Widgets that represent the hard keys on this cellphone have been selected and, from the Modify menu, the designer has chosen Set Frame Template. This is indicated in the status message at the bottom left of the window, which reads: “Frame Template created”. Each time a new Frame is created in this Design, it will already have these buttons in it.

To clear the Frame Template so your new Frames can be created empty,

- From the Modify menu, choose Clear Frame Template.

4.2.5 Selecting Frames

In the Design window, click on a Frame to select it. It will turn bright blue to indicate selection.

To select multiple Frames either

- Click in the background of the Design and drag a bounding box around the Frames you want to select.

OR

- Shift-click each Frame separately.

4.2.6 Renaming a Frame

Mac: ⌘R

Win: CTRL+R

To change the name of a Frame, do one of the following:

- In the Design window, select the Frame.

- From the Edit menu, choose **Rename Frame**.

OR

- Double-click the Frame's name. The Frame title bar turns light blue when the name can be changed.



A Frame name must be unique within a Design.



Text in CogTool's item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

4.2.7 Moving and Aligning Frames

You may arrange the Frames to make the Design more visually pleasing and easier to follow.

To move a Frame, simply click on the center of the Frame and drag it to a new location.

Mac: ⌘+ARROWS

Win:

CTRL+ARROWS

To move a Frame in fractional increments, use **Nudge**:

- Select any number of Frames
- From the **Modify** menu, choose **Nudge**.
- Select the desired direction.

OR

- While holding the ⌘ key (Mac) or the Control key (Win), use the arrow keys on the keyboard to move the Frame.



To reduce confusion, move the Frames apart so they are not overlapping or stacked.

To align Frames:

- Select two or more Frames
- From the **Modify** menu, choose Frame **Alignment**.
- Select the desired alignment.

4.2.8 Cutting, Copying, and Pasting a Frame

To cut one or more Frames:

- Select the Frames.
- From the **Edit** menu, choose **Cut Frame**.

Mac: ⌘C

Win: CTRL+C

The Frames will disappear and be put on the clipboard, ready to be pasted.



When you copy a Frame, only the contents of the Frame are copied; Transitions are not. If you want to preserve Transitions,

use the **Duplicate** command.

To copy one or more Frames:

Mac: ⌘X
Win: CTRL+X

- Select the Frames.
- From the **Edit** menu, choose **Copy**.
The Frames will be put on the clipboard, ready to be pasted.

To paste a Frame into a Design.

Mac: ⌘V
Win: CTRL+V

- Open the target Design window.
- From the **Edit** menu, choose **Paste Frame**.
This places the Frames on the clipboard into the Design. If the name of the pasted Frame already exists in the Design, the pasted Frame is given a unique suffix (e.g. “LastFrame[1]”).



You can paste frames into any Design. You can paste frames into the Design you just copied them from (similar to **Duplicate**), into a different Design in the same Project, or into a Design in a different Project.

4.2.9 Deleting a Frame

To delete one or more Frames:

Mac: ⌘⌫
Win: CTRL+DELETE

- Select the Frames.
- From the **Edit** menu, choose **Delete Frame**.
- You will be asked to confirm the deletion. If you click **OK**, then the Frames will disappear.



If you have already demonstrated your Design, you will have to redo your Demonstration after deleting Frames that are used in the Demonstration.



Deleting a Frame has the same effect as cutting a Frame, except that the Frame is not on the clipboard and cannot then be pasted.

4.2.10 Duplicating a Frame

When you copy a Frame, you only copy the content held within the Frame. No Transitions to other Frames are copied. To make an exact duplicate of a Frame, including Transitions, use the **Duplicate** command instead of **Copy**.

To duplicate one or more Frames:

Mac: ⌘D
Win: CTRL+D

- Select the Frames.
- From the **Edit** menu, choose **Duplicate Frame**.
This creates new Frames slightly down and to the right of the original Frames,

named with unique suffixes (e.g. “LastFrame[1]”).

OR

- Select the Frames.
- Hold down the Option key (Mac) or Control key (Windows), click in a selected Frame and drag it to a new position.
This creates new Frames at the point you released the mouse button, named with unique suffixes (e.g. “LastFrame[1]”).

4.2.11 Editing a Frame

The content of a Frame represents what the user will see and hear as they operate the proposed system. You must construct that content by editing Frames.

To edit a Frame, do one of the following:

Mac: ⌘E

Win: CTRL+E

- Select the Frame.
- From the **Edit** menu, choose **Edit Frame**.

OR

- Double-click the body of the Frame.

4.2.12 Zooming a Frame

When in a Frame window, you may zoom in and out on a Frame to make working with it easier. This is often necessary when using a large background image or when your Frame has many Widgets.

Zoom In

Mac: ⌘=

Win: CTRL+=

To zoom a frame, do one of the following:

- From the Modify menu, choose the Zoom option you want.

OR

- Choose your zoom settings from the box at the bottom of the Frame window.

Zoom Out

Mac: ⌘-

Win: CTRL+-



Normal Zoom makes the contents of the Frame revert to the default size. **Zoom to Fit** makes the contents of a Frame fit the available space in the window.

Normal Zoom

Mac: ⌘0

Win: CTRL+0

Zoom to Fit

Mac: ⌘/

Win: CTRL+/

4.2.13 Setting the Background Image of a Frame

You may use a background image for your Frame. Background images are useful in helping you to build an interface based on Design drawings or screen captures of an existing interface.

You can set a Frame’s background image any time that Frame is selected in the Design

window or the Frame's window is active.

From the Design window:

- Select the Frame.
- From the **Modify** menu, choose **Set Background Image**.

OR

From the Frame window:

- From the **Modify** menu, choose **Set Background Image**.

Then choose the image file from the **Open File** dialog box. JPG, PNG, and GIF are acceptable file formats.



On Windows, you can also paste images from the clipboard into your Frame.

Mac: ⌘B

Win: CTRL+B

4.2.14 Removing the Background Image of a Frame

From the Design window:

- Select the Frame.
- From the **Modify** menu, choose **Remove Background Image**.

OR

From the Frame window:

- From the **Modify** menu, choose **Remove Background Image**.

Mac: ⌘B

Win:

CTRL+SHIFT+B

4.2.15 Bringing a Frame to the Foreground

To bring your open Frame to the foreground:

- From the **Window** menu, choose the desired **Project**.
- From the drop-down menu, choose the **Frame**.

4.2.16 Closing the Frame Window

To close a Frame:

- From the **File** menu, choose **Close**.

Mac: ⌘W

Win: CTRL+W

4.3 Widgets

4.3.1 What is a Widget?

A Widget is an element on a Frame with which a user may interact, such a button, check boxes, a hierarchical menu, etc.. You will populate your Frames with Widgets as needed to express your Design.

There is no limit to the number of Widgets that can be included in a Frame, but you may not need to put Widgets in for every interactive element in the Design. If you are using CogTool to make predictions of task execution time for skilled users, the underlying human performance model only needs the widgets that are actually used in the tasks you are investigating. If you are using CogTool to create HTML to share with a development team, they may also need only a fraction of the widgets that might appear on every Frame. Depending on your needs, your Designs may have few Frames, sparsely populated with Widgets, even to depict complex interfaces.

For example, the Frame shown below (see Figure 4-13), represents the first screen of a NYC Guide for a PalmVx. One task that the UI designer wants to investigate both with CogTool's human performance models and with user testing is to find the open hours of the Metropolitan Museum of Art. Only three buttons on the first screen are on a reasonable path to this goal, Maps, All Places, and Museums, so the UI designer need only place Widgets for those three buttons and leave all the rest unspecified until other tasks become important in the investigation.

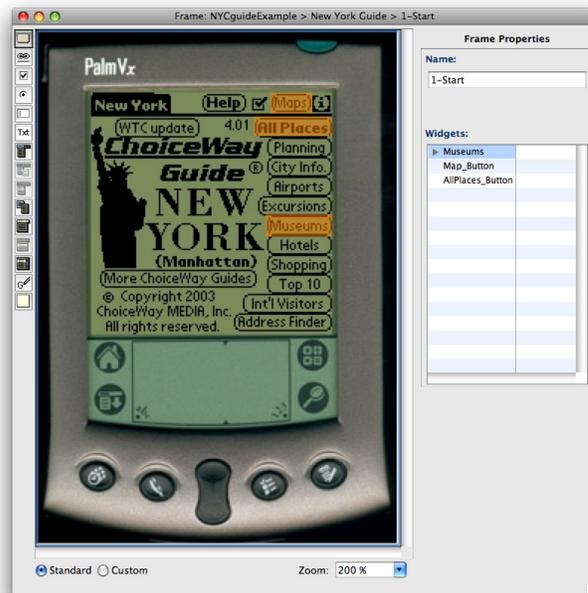


Figure 4-13: This Frame canvas shows a background image of the PalmVx running the ChoiceWay New York City Guide . Three standard button widgets have been defined (in orange, a settable color). The Frame Properties pane lists those three buttons.

Widgets will be the source of Transitions in the Design window. Depending on its type, a Widget can support different Transitions that represent different user actions on that Widget. Transition link to a new Frame or the same Frame to represent the systems' response to a user's action (see **Section 4.4 Transitions** for more information).

4.3.2 Types of Widgets

Choosing appropriate types of Widgets is very important when populating your Frames. The type of Widget determines the types of actions that can be represented with Transitions and greatly influences results of CogTool's predictive human performance modeling. However, choosing appropriate Widgets is in line with the job of a UI designer, so what you do in your normal design and prototyping work fits right into CogTool's process, too. First decide whether Standard widgets are sufficient for your design or whether you must use Custom widgets; then decide which specific widget to use.



A Widget cannot be changed to a different type once it is created.

4.3.2.1 Standard or Custom Mode?

CogTool provides two distinct modes for creating Widgets, Standard and Custom.

Following the HCI maxim “Make frequent tasks easy and infrequent tasks possible,” Standard mode makes it very easy to create complex interactive widgets. For example, if you choose to create a menu in Standard mode, the entire hierarchical menu structure can simply be typed in on the keyboard and it will open and close appropriately when you interact with it. Type it in once and paste the whole structure into the top of each Frame. Likewise, if you create a radio button in Standard mode, you type in all the names in a system of radio buttons and they automatically perform so that only one can be selected at a time. Widgets created in Standard mode can be rendered as wire frames, with a Macintosh skin or with a Windows XP skin. You can easily build up entire interfaces using rendered Widgets on a blank Frame canvas. We expect that Standard mode will make life vastly easier when prototyping a majority of interfaces.

Custom mode allows you to define non-standard Widgets that look and interact in any way you wish, but with more effort and care than Standard mode requires. Widgets created in Custom mode cannot be connected into systems like in Standard mode, so you have to build them separately, and combine them carefully so you get proper behavior and, more importantly, valid human performance models. For example, if you wanted to build a pie menu, you would have to include background images that look as you want the menu to look in all its expansions and your Design would have to ensure (through its system of Frames and Transitions) that a submenu can never be clicked on until the high-level menu that contains it is expanded.

Previous versions of CogTool did not have Standard mode so all complex systems of Widgets had to be built up as they are in Custom mode today. An analysis of errors made by CogTool users revealed that this process was the most error-prone part of

using CogTool. Hence, we introduce Standard mode to reduce the possibility of user error, simplify both the creation and debugging of CogTool prototypes, and increase the accuracy of CogTool's predictions. If it fits your design needs, we recommend using Standard mode.



A Widget cannot be changed to a different type once it is created.

4.3.2.2 Choosing Widget Types

Because choosing Widget types is so important and can't be changed, even experienced CogTool users refer to documentation, especially if using Custom mode. First decide whether Standard widgets are sufficient or whether you must use Custom widgets; then decide which specific widget to use. To make the necessary information accessible, which have put a table of Widget types, their meanings, and some examples about when to use each type in an Appendix. Please see **Appendix B: Types of Widgets and When to Use Them** for this decision aid. After you have chosen the type of Widget you need in a Frame, refer to the sections below to create and manipulate those Widgets.

4.3.2.3 A Road map of the Widgets Section

To reduce redundancy, we have grouped Widgets into subsections of this document. The first subsection presents all the information that is common to all widgets. The next section is Simple Widgets, where we discuss Links, Graffiti and Non-Interactive Widgets and the actions that are common to all Widgets (e.g., cut/copy/paste). The next section is Moded Widgets, where we discuss Buttons and Check Boxes. Finally, we present each of the more complex Widgets in their own sections in increasing complexity: Radio Buttons, Pull-down Lists, Menus and Context Menus, and Text Boxes and Text.

4.3.3 Common Actions on Widgets

4.3.3.1 Creating a New Widget

After you have decided which Widget to use and whether your interface will require standard or custom mode, you can create your Widget.

To create a Widget

- Select the Standard or Custom radio button at the bottom of the Frame window.
- From the toolbar, choose the type of Widget you want to create
- Drag a rectangle across an area of the open Frame.

OR

- From the **Create** menu, choose **New Widget**.

Mac: ⌘I

Win: CTRL+SHIFT+I



When you use the New Widget command from the menu or short-cut keys, a square Widget will be placed in the upper left corner of the Frame. You can move and resize the Widget as appropriate for your interface.



It is much more efficient to select a Widget tool and then draw it on the canvas where you want it instead of using the **New Widget** command. (We included the New Widget command only for discoverability and compatibility with other similar systems; nobody we know uses it!)



When you select the Standard or Custom radio button, it stays selected for all new Widgets until you change it, so you only have to select it once for you entire Design if you do not need to mix modes. Standard is selected by default.

The appearance of a new Widget is determined by its type, so please see the subsequent sections for the specific appearance and next steps for creating different types of Widgets.



When you have the same Widget on different Frames in a Design, prepare that widget completely in one Frame and then copy and paste it into all other Frames that need it. This will save you effort and also make CogTool's predictions more accurate because CogTool will know that it is exactly the same Widget with the same name, size and position on each Frame.



Widgets can only be rectangles. This is an approximation that is sufficient for making predictions of human performance..

4.3.3.2 Naming a Widget and Entering its Display Label

Every Widget has a Name and an optional Display Label. The Name must be unique to a Frame and is used internally by CogTool. The Display Label is the label that a user would see on the display of the system you are prototyping. There is *always* a Name; there is only a Display Label if you want to have one.

You would want to use Display Labels if you are working from a blank canvas because the Display Label is shown on the canvas, but Names are not. For the same reason, you would not want to use Display Labels if you are using a Frame background image that has pictures of widgets on it that already have labels because the Display Labels will be superimposed on the background. CogTool predictions of skilled performance time do not need a Display Label to run correctly, so you may elect to ignore them if you have a background image that already contains labels.

When you create a Widget, an editable text box comes up on the Widget ready to receive the Display Label and a default name is inserted into the **Widget Name** field, located in the Widget Properties to the right of the Frame. You may use alphanumeric characters, spaces, dashes, and underscores to name your Widget or in its Display Label

To Rename a Widget: You can always change a Name or Display Label by selecting the widget and changing the text in Widget Name or Display Label fields in the Properties pane. (see Figure 4-14) You can also change the Display Label by double-clicking on the Widget, which brings up an editable text box.

Mac: ⌘R

Win: CTRL+R



When not using Display Labels, we suggest that you change the Name default to a more meaningful name so that when it appears in elsewhere in the system or in exported files, you can associate it with something more meaningful than “Widget 1.”



You cannot change the size, type, or appearance of the font in the Display Label. The underlying cognitive model gives valid predictions of the performance time of skilled users no matter what the font looks like (just as skilled human users would have learned the procedures no matter what the font).

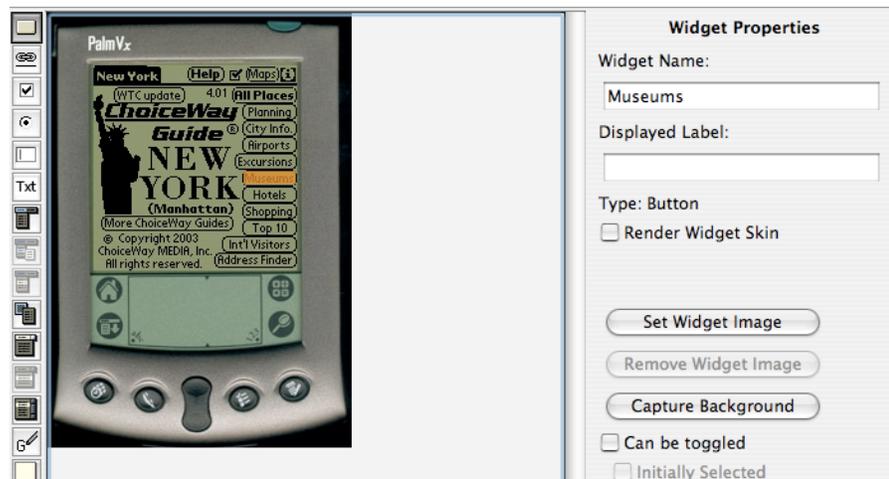


Figure 4-14: This Widget has been named “Museums.”



Some words that are commonly used in Button Names and Display Labels have special meaning to CogTool. They signal termination of a Task, which has been shown by prior psychology research to effect people’s behavior. These words are: Cancel, Yes, No, Exit, Abort, Quit, Save, Done, and OK. If either the Widget’s Name or Display Label is exactly one of these words (not case sensitive), CogTool will make different human performance predictions than if it is something slightly different (e.g., “OK-button” instead of “OK”). Be sure to use one of these exact words as the button’s Name or Display Label if it does indeed function as the terminator of a task.



Text in CogTool’s item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

4.3.3.3 Selecting Widgets

In the Frame window, click on a Widget to select it. It will turn a different color to indicate selection and a gray bounding box will appear surrounding the Widget.

To select multiple Widgets either

- Shift-click in the background of the Frame and drag a bounding box around the Widgets you want to select.

OR

- Shift-click each Widget separately.

You can Select-All Widgets from the **Edit** menu by choosing **Select All Widgets**, or ⌘A (Mac) or CTRL+A (Windows).

4.3.3.4 Setting a Widget's Appearance

4.3.3.4.1 Setting a Widget Layer Color

The Widget layer color is the transparent color that fills each Widget so you can see it as you work with it; its complement in the color wheel is the highlight color indicating selection. The default color is orange because that is unlikely to blend in with most background images. If you prefer another color, or if orange or its complement make working with your background image difficult, you can change it. The color is set per Frame.

To change the color of all the Widgets in a Frame:

Mac: ⌘⌘C

Win:

CTRL+SHIFT+C

- From the **Modify** menu, choose **Set Widget Layer Color**.
A color-picker appears.
- Choose the color you want from the color-picker.
All widgets in that frame change to the new color.



The Widget Layer color has no bearing on the appearance of Widgets when they are exported to HTML.

4.3.3.4.2 Rendering a Widget Skin

A Widget skin can be rendered to resemble objects, such as buttons, on different systems (Mac OS X or Windows XP).

To render a Widget skin:

- Select the Widget.
- In the Widget Properties pane, check the **Render Widget Skin** check box.
- From the **Modify** menu, choose **Set Design's Widget Skin**.



Rendering is done per Widget, but selecting the skin is done per Design. When you set the skin, it will change all rendered Widgets in the whole Design.



You can Render all the Widgets in a Design, or Un-render all Widgets in a Design at once, using the **Render All Design's Widgets Skin** or **Un-render All Design's Widgets** commands in the Modify menu.

The available Widget skins are as follows:

Wireframe: Each Widget is outlined in black. Each Widget retains the color of the Widget layer and continues to be transparent so that the background image (if any) shows through.

Mac OS X & Windows XP: Each Widget has the appearance of the corresponding interactive Widget in the specified OS. These skins are opaque, so the background image (if any) does not show through.



CogTool's skins may not be as professional as you would like for the Mac OS X and Windows XP. Try them out to see if you like them. You can always capture the background of any Widget from a screen shot to get the look you want.

4.3.3.4.3 Setting and Removing an Image for a Widget

Widgets can have an image of their own. This is useful for creating a new layout that is not based on an existing system or background image. It is especially useful for creating custom Widgets that look different from anything seen before.

To set an image for a Widget:

- Select the Widget.
- From the **Modify** menu, select **Set Widget Image**.
- From the standard Open File Dialog box, choose your file.



Only JPG, PNG, or GIF are acceptable formats for Widget images.

You may also choose to have the Widget match the background image of the Frame beneath it. You will then be able to maintain a consistent image for the Widget as you move it around the Frame (see Figure 4-15).

To capture the image from the Frame's background:

- Select the Widget.
- From the **Modify** menu, select **Capture Widget Image**.



Figure 4-15: The Museum button has been captured for the Widget from the upper right of the Frame background. The Widget has been moved to the Graffiti area (a silly thing to do) to illustrate how the captured image looks.



You can use this feature to create entirely new layouts from a screen shot of a Frame. Create a Frame with the desired background and place Widgets over all the widgets that appear in the background. Capture the background for each Widget. You can then remove the Frame background and move the Widgets around, or copy and paste the Widgets into a new blank or neutral background.

To remove the image from a Widget:

- Select the Widget.
- From the **Modify** menu, select **Remove Widget Image**.



Removing the Widget image will not remove the background image on the Frame, if there is one.

4.3.3.5 Moving and Aligning Widgets

Most Widgets can be moved by selecting them, then clicking and dragging them by their body (within the selection bounding box). Complex standard Widgets with many parts have to be dragged by the bounding box itself, because dragging the interior space moves the pieces of the Widget relative to its other pieces, e.g., changing the order of menu items. These movement procedures will be discussed in the sections about those Widgets.

You can always move one or more Widgets by “nudging” them. To nudge one or more selected Widgets, do one of the following:

- Select any number of Widgets
- From the **Modify** menu, choose **Nudge**.
- Select the desired direction.

OR

- While holding the ⌘ key (Mac) or the Control key (Win), use the arrow keys on the keyboard to move the widget.

You can align multiple Widgets precisely on the Frame background.

To align Widgets:

- Select two or more Widgets

Mac: ⌘+ARROWS

Win:

CTRL+ARROWS

- From the **Modify** menu, choose Widget **Alignment**.
- Select the desired alignment.

4.3.3.6 Layering Widgets

In CogTool, Widgets can be drawn on top of each other, therefore, they can partially or fully occlude other Widgets. You can set the order of this layering with the standard drawing program commands to “Bring to Front,” “Bring Forward,” “Send Backward,” and Send to Back.”

To layer Widgets:

- Select one or more Widgets.
- From the **Modify** menu, choose **Widget Layering**, then chose the desired layering command.



Drawing overlapping Widgets has implications for defining Transitions and demonstrating tasks (see **Chapter 5: Quantitative Analysis** for more information about).

4.3.3.7 Resizing Widgets

To resize a Widget:

- Select the Widget.
- Resize by clicking-and-dragging in the small white boxes at the corner of the Widget.



Resizing of complex Widgets may be more complex; it will be discussed in the sections about those types of Widget.

4.3.3.8 Cutting, Copying and Pasting Widgets

In CogTool, you can copy or duplicate a Widget.

To cut one or more Widgets:

- Select the Widgets.
- From the **Edit** menu, choose **Cut Widget**.
- You will be asked to confirm the deletion. If you click **Yes**, the Widgets will disappear and be put on the clipboard, ready to be pasted.

Mac: ⌘C

Win: CTRL+C

To copy one or more Widgets:

- Select the Widgets.
- From the **Edit** menu, choose **Copy Widgets**.
The Widgets will be put on the clipboard, ready to be pasted.

Mac: ⌘X

Win: CTRL+X



When you cut or copy a Widget, only the contents of the Widget are copied; Transitions from that Widget are not. If you want to preserve Transitions, use the **Duplicate** command.

To paste a Widget into a Frame.

Mac: ⌘V

Win: CTRL+V

- Open the target Frame window.
- From the **Edit** menu, choose **Paste**.

This places the Widgets on the clipboard into the Frame. If the name of the pasted Widget already exists in the Frame, the pasted Widget is given a unique suffix (e.g. “LastWidget[1]”).



When you paste a Widget into the same Frame, it is pasted directly on top of the original Widget and is given a new unique suffix (“MyWidget[1]”). Move the newly pasted Widget off of the original to see both Widgets.



You can paste Widgets into any Frame. You can paste Widgets into the Frame you just copied them from (similar to **Duplicate**), into a different Frame in the same Design, or into a Frame in a different Design or Project.



If the same interface Widget appears in different Frames of the Design, it is important that the Widgets be placed and sized identically. This is imperative to ensure the correct prediction of human performance when there are several actions in a row on the same Widget. To make sure that all Widgets are identical, you should create the Widget on one Frame and copy and paste it into all the other Frames. Not only is this more efficient, it ensures higher accuracy of performance predictions.

4.3.3.9 Deleting Widgets

To delete one or more Widget:

Mac: ⌘⌫

Win:

CTRL+DELETE

- Select one or more Widgets
- From the **Edit** menu, choose **Delete Widget**.
- You will be asked to confirm the deletion. If you click **OK**, the Widgets will disappear.



Deleting a Widget has the same effect as cutting a Widget, except that the Widget is not on the clipboard and cannot then be pasted.

4.3.3.10 Duplicating a Widget

When you copy a Widget, you only copy the content held within the Widget. No Transitions to other Frames are copied. To make an exact duplicate of a Widgets, including Transitions, use the **Duplicate** command instead of **Copy**.

To duplicate one or more Widget:

Mac: ⌘D

Win: CTRL+D

- Select the Widgets.
- From the **Edit** menu, choose **Duplicate Widget**.

This creates new Widgets slightly down and to the right of the original Widgets,

with the same Display Labels, but named with unique suffixes (e.g. “LastWidget[1]”).

OR

- Select the Widgets.
- Hold down the Option key (Mac) or Control key (Windows), click in a selected Widget and drag it to a new position.

This creates new Widgets at the point you released the mouse button, with the same Display Labels, but named with unique suffixes (e.g. “LastFrame[1]”).



When you duplicate a Widget, an identically sized Widget appears to the lower right of the original Widget and is given a new unique suffix.



Text in CogTool’s item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

4.3.3.11 Creating a “Template” of Widgets

As discussed in the section on Creating Frames, when you are prototyping a system that has many of the same widgets in multiple Frames, e.g., a cellphone with many hard buttons or a airline cockpit device like the Control Display Unit where only the screen changes from Frame to Frame, CogTool provides an efficient way for you to create a template of widgets that will automatically populate each new Frame created in the Design.

To create a Frame Template of Widgets

- In any Frame window, select the Widgets you want to be on every new Frame in the Design.
- From the Modify menu, choose Set Frame Template.
- Create new Frames, and each of them will be pre-populated with the Widgets in the template.



Figure 4-16: All the Widgets that represent the hard keys on this cellphone have been selected and, from the Modify menu, the designer has chosen Set Frame Template. This is indicated in the status message at the bottom left of the window, which reads: “Frame Template created”. Each time a new Frame is created in this Design, it will already have these buttons in it.

To clear the Frame Template so your new Frames can be created empty,

- From the Modify menu, choose Clear Frame Template.

4.3.4 Simple Widgets: Links, Graffiti®, and Non-Interactive

 Link Widget tool

 Graffiti® Widget tool

 Non-interactive Widget tool

Links, Graffiti®, and Non-interactive Widgets are the simplest type of Widgets. Their Standard mode of creation and their Custom mode of creation are the same, and the only settable properties are their name, their display label, whether they are rendered, and whether they have a background image. All the common actions on Widgets work on these Widgets as described in the previous section.

The use of the Display Label is the only factor that requires some thought when using these Widgets

- Link Widgets usually use the Display Label as the text of the link. When a background image is used, you may not want to use a Display Label so the image of the link on the background image shows through.
- Graffiti® widgets usually doesn't use a Display Label because any text typed into the Display Label field will show up in the Graffiti® Widget and the Graffiti® area on a UI is usually blank.
- a Non-interactive Widget represents a place on the display that users deliberately look at to see and comprehend information in the course of doing their work, but they do not click on or otherwise interact with that information. Non-interactive Widgets often use their Display Labels to present that information, or they are placed on top of a background image with no Display Label, or they have a Widget image and again no Display Label.

4.3.5 Moded Widgets: Buttons and Checkboxes

 Button Widget tool

 Checkbox Widget tool

Buttons and Checkboxes are moded Widgets, in that, when they are created as Standard Widgets, they can be in one of two states. Buttons can be defined so they can be toggled (selected and not-selected); Checkboxes can be checked or unchecked. The initial state of these moded Widgets must be set so that it is in the correct state when the Design enters the Frame. This state is set in the Widget properties pane.

Except for setting state, Buttons and Checkboxes respond to all the common actions on Widgets, as described in a previous section.



Custom Buttons and Checkboxes do not have state. To simulate state with custom Buttons and Checkboxes, you would have to have different images (one selected and one un-selected) on different Frames, with click transitions to move between those Frames. Standard Buttons and Checkboxes do this behavior for you.

4.3.5.1 Buttons

 Button Widget tool

After creating a Button Widget in standard mode, its Widget properties pane will appear. At the bottom of this pane is a check box determining whether this button should toggle or not. If you check this box, then the check box for whether it should be ini-

tially selected or not can be checked.

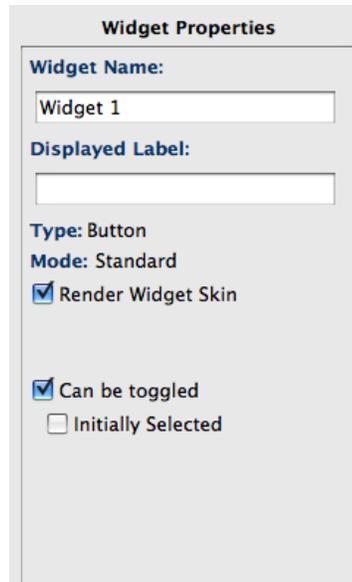


Figure 4-17: The Widget properties pane where Button Widgets can be made toggleable and but not initially selected.

When rendered in one of the operating system skins, the Button will look different when it is selected or not selected. (It will not look different rendered in wire frame or when not rendered.)

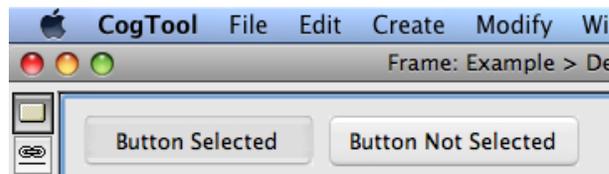


Figure 4-18: Two toggleable buttons, rendered with the Mac OSX skin. The one on the left is selected; the one on the right is not. When selected, the button appears depressed, whereas the un-selected button appears slightly raised.

Some words that are commonly used in Button Names and Display Labels have special meaning to CogTool. They signal termination of a Task, which has been shown by prior psychology research to effect people's behavior. These words are: Cancel, Yes, No, Exit, Abort, Quit, Save, Done, and OK. If either the Widget's Name or Display Label is exactly one of these words (not case sensitive), CogTool will make different human performance predictions than if it is something slightly different (e.g., "OK-button" instead of "OK"). Be sure to use one of these exact words as the button's Name or Display Label if it does indeed function as the terminator of a task.

4.3.5.2 Checkboxes



Checkbox Widget tool

Checkbox Widgets are always moded; they can be checked (selected) or unchecked (not selected). In the Widgets property pane of a Checkbox Widget, set the initial state of the Widget with the Inially-Selected chackbox at the bottom of the pane.

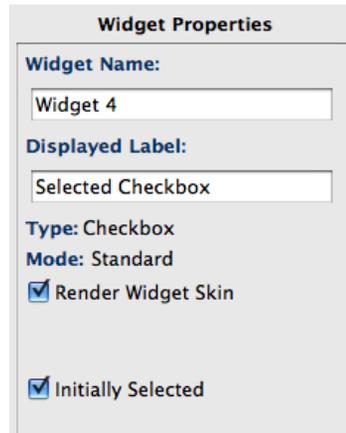


Figure 4-19: Checkbox Widgets can be initially selected (checked).

The biggest advantages of using standard Checkboxes is realized when you render them and use Display Labels. When rendered, a box is put to the left of the Display Label to act as the Checkbox. A rendered Checkbox will look different when it is selected or not selected. It will have a check or X in it when selected, depending on which skin is used for rendering.

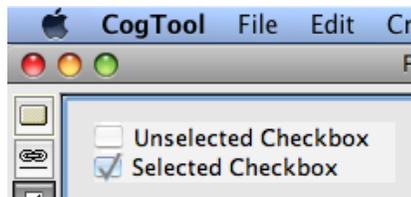


Figure 4-20: Two Checkboxes, rendered with the Mac OSX skin. The one on the top is not selected; the one on the bottom is.

4.3.5.2.1 Creating Checkbox Widgets

In many interfaces, Checkboxes are aligned vertically in one or more columns.. Therefore, CogTool makes it easy to create a set of aligned Checkbox Widgets.

To create a set of Checkbox Widgets.

- From the toolbar, select the Checkbox tool 
- Drag a rectangle across an area of the open Frame large enough to contain the Checkbox square and the longest Display Label in the set of Checkboxes.



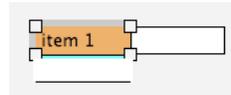
At this point, you are not sizing the area that will be occupied by the entire set of checkboxes; You are sizing only the single Checkbox that has the longest Display Label.

- A Widget appear that has one highlighted Widget (in orange, below) with its

Display Label ready to receive typing (white box). Type the Display Label for the first checkbox and it will appear in the white box that appears on the Widget.



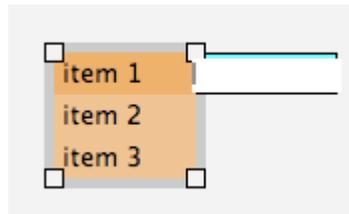
- Pressing the Enter key will create the first Checkbox and move the cursor to the Checkbox below the first.



- Continue to label the Checkboxes by typing in the white box and hitting Enter to get to the next Checkbox. This creates a column of Checkboxes.

FYI

Checkboxes can be a column of items or a grid of items comprised of rows and columns. To create a second column of items, navigate to the Checkbox Widget in the top of the column using the arrow keys (or clicking on it), then press the right arrow key on the keyboard, and type the label in the white box that appears to the right of the top Checkbox Widget.



- The checkbox image is inserted into the Checkbox Widget when the Widget is rendered (check **Render Widget Skin** in the Widget properties pane).

FYI

The checkbox image size is determined by the size of the checkbox Widget. The size of the checkbox image itself can not be changed manually. However, you can use a background button image to get a box of the desired size if you like.

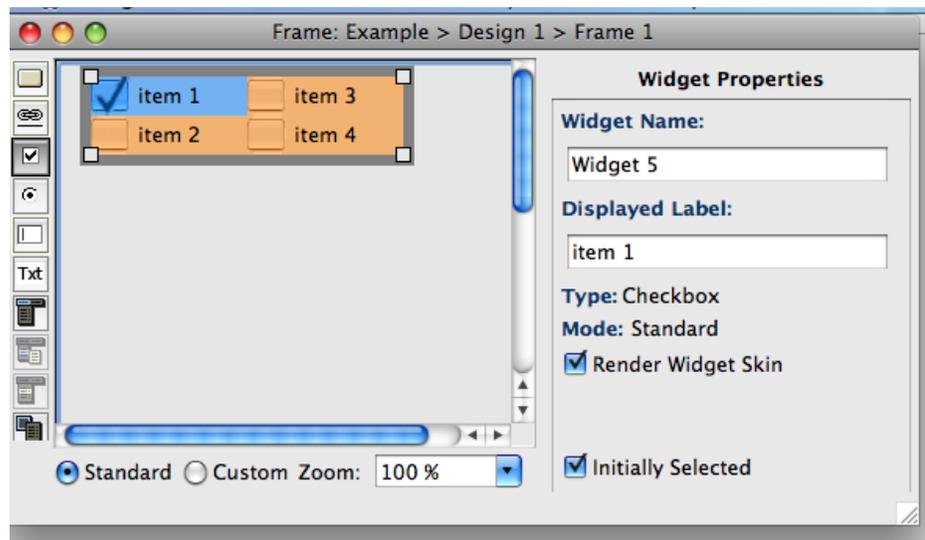


Figure 4-21 The Frame window with a set of Checkbox Widgets. The **Render Widget Skin** checkbox must be selected for the boxes to appear on the Widget (this example has the Mac skin set for the design). The first checkbox is initially selected, because that box is selected at the bottom of the Widget Properties pane.

fix You cannot reorder the individual Checkbox Widgets after you have created the set. If you want to change the order of the Checkbox Widgets, you will have to retype the information in the Displayed Label field on the Widget or in the Widget Properties pane.

4.3.5.2.2 Arranging Checkbox Widgets

The grid on which the Checkbox Widgets are laid out can be manipulated to fill the space you wish it to occupy. You can have different amounts of space between subsets of checkboxes. This allows you to make room between sets of checkboxes for other widgets that are enabled only when a checkbox is selected (e.g., a set of radio buttons or a set of check boxes).

- Click on the checkbox above or to the left of which you want to insert white space. A bold line appears on the top and, if there are checkboxes to the left, at the left of this Widget.
- Grab the line and move it to wherever you want the Widgets to be placed.

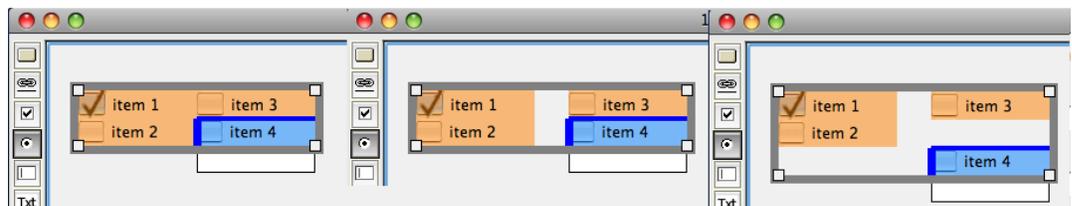


Figure 4-22: A progression of checkboxes being arranged. At the left, the user selected the Widget called “item 4”, which highlighted it and showed the bold blue lines on the top and left. The user grabs the left blue line and moves it to the right, moving the

column of buttons as shown in the center.. The user then grabs the top blue line and moves it down, resulting in the arrangement on the right.

4.3.6 Radio Buttons: A Set of Widgets Working Together

Radio Button Widget tool

Standard Radio Buttons are created as a set and CogTool enforces that only one of them can be selected at any time. (If you create Radio Buttons in custom mode, you must manage that behavior yourself with multiple Frames and Transitions between them.)

Except for creating and arranging, Radio Button Widgets respond to all the common actions on Widgets, as described in a previous section.

The biggest advantages of using standard Radio Button Widgets is realized when you render them and use Display Labels. When rendered, a circle is put to the left of the Display Label to act as the Radio Button. A rendered Radio Button will look different when it is selected or not selected. It will have a filled circle when selected, depending on which skin is used for rendering.

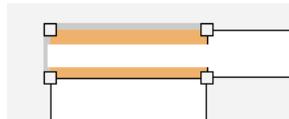
4.3.6.1 Creating Radio Button Widgets

To create a set of Radio Button Widgets:

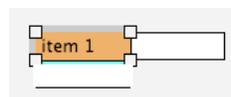
- From the toolbar, select the Radio Button tool 
- Drag a rectangle across an area of the open Frame large enough to contain the Radio Button circle and the longest Display Label in the set of Radio Buttons.

 At this point, you are not sizing the area that will be occupied by the entire set of radio buttons,; You are sizing only a single Radio Button that has the longest Display Label.

- A Widget appear that has one highlighted Widget (in orange, below) with its Display Label ready to receive typing (white box). Type the Display Label for the first radio button and it will appear in the white box that appears on the Widget.

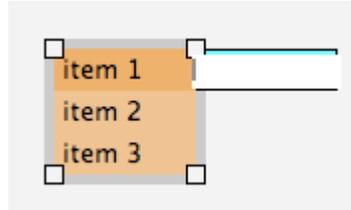


- Pressing the Enter key will create the first Radio Button and move the cursor to the Radio Button below the first.



- Continue to label the Radio Buttons by typing in the white box and hitting Enter to get to the next Radio Button. This creates a column of Radio Buttons.

- FYI** Radio buttons can be a column of items or a grid of items comprised of rows and columns. To create a second column of items, navigate to the radio button Widget in the top of the column using the arrow keys (or clicking on it), then press the right arrow key on the keyboard, and type the label in the white box that appears to the right of the top Radio Button Widget.



- The radio button image is inserted into the radio button Widget when the Widget is rendered (check **Render Widget Skin** in the Widget properties pane).

- FYI** The radio button image size is determined by the size of the radio button Widget. The size of the radio button image itself can not be changed manually. However, you can use a background button image to get a radio button of the desired size.

- You can specify which Radio Button is initially selected by choosing that item from the **Initially Selected Item** pull-down list on the property pane of any of the Radio Buttons in the set.

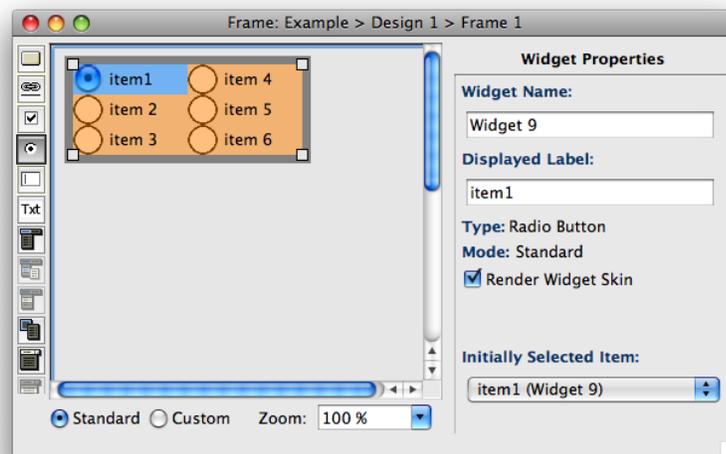


Figure 4-23: The Frame window with a set of radio button Widgets, rendered with the Macintosh skin. The **Render Widget Skin** checkbox must be selected for the radio buttons to appear on the Widget. Item 1 is the **Initially Selected Item**.

- FYI** You cannot reorder the individual radio button Widgets after you have created the set. If you want to change the order of the buttons, you will have to retype the information in the Displayed Label field on the Widget or in the Widget Properties pane.

4.3.6.2 Arranging Radio Buttons

The grid on which the Radio Buttons are laid out can be manipulated to fill the space you wish it to occupy. You can have different amounts of space between subsets of Radio Buttons. This allows you to make room between sets of radio buttons for other widgets that are enabled only when a radio button is selected (e.g., another set of radio buttons or set of check boxes).

- Click on the radio button above or to the left of which you want to insert white space. A bold line appears on the top and, if there are Radio buttons to the left, at the left of this Widget.
- Grab the line and move it to wherever you want the Widgets to be placed.

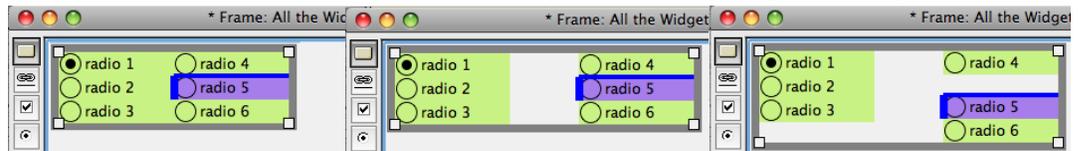


Figure 4-24: A progression of Radio Buttons being arranged. At the left, the user selected the Widget called “radio 5”, which highlighted it and showed the bold blue lines on the top and left. The user grabs the left blue line and moves it to the right, moving the column of buttons as shown in the center. The user then grabs the top blue line and moves it down, resulting in the arrangement on the right.

4.3.7 List Boxes: Widgets Combined to Make a Whole

List Box Widget tool

A standard List Box is created as a set of items combined into one List Box that is automatically aligned and can move together as one Widget. CogTool does not yet have the capability of making List Box Widgets behave like real List Boxes (e.g., a large list cannot extend beyond a window, they cannot scroll, etc.), but the ease of entering, re-arranging, aligning and moving as a whole is better than making separate custom List Boxes and manipulating them separately yourself.

Except for creating, changing the order of items, and moving, List Box Widgets respond to all the common actions on Widgets, as described in a previous section.

The biggest advantages of using standard List Box Widgets is realized when you use Display Labels.

4.3.7.1 Creating a List Box

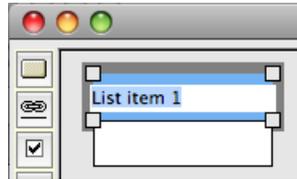
To create a list box:

- Select the list box icon  from the Widget toolbar.
- Drag a rectangle across an area of the Frame large enough to contain the longest Display Label in list.

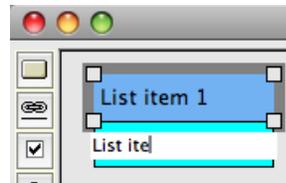


At this point, you are not sizing the area that will be occupied by the entire list; You are sizing only a single List Box item that has the longest Display Label.

- A Widget appear that has one highlighted Widget (in orange, below) with its Display Label ready to receive typing (white box). Type the Display Label for the first item and it will appear in the white box that appears on the Widget.



- Pressing the Enter key will create the first list item and move the cursor to the list item below the first.



- Continue to label the items by typing in the white box and hitting Enter to get to the next item, until all the items in that part of the List Box that will be visible in our Design in this Frame are entered.



If the real List Box you are trying to represent has more items in it than are visible at any one time, you can simulate that by having only the first visible lines in the List Box on the first Frame, put a custom Button Widget that looks like a down-scroll arrow next to the bottom of the List Box Widget. This button can transition to another Frame that has a List Box Widget with the next set of items visible in it. This Frame can have a custom Button Widget than looks like an up-scroll arrow next to the List Box Widget, that transitions back to the previous Frame. You can simulate any length List Box through a series of Frames in this way.

4.3.7.2 Rearranging List Box Items

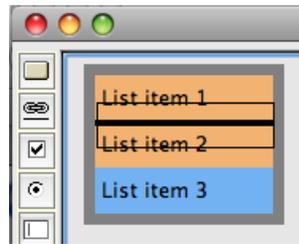
To change the order of a List Box item:

- Click the List Box item to select it.
- Drag the List Box item to a new location within the same List Box Widget or to another List Box Widget.



When you drag a Widget, an outline of the Widget and a bold black position indicator will appear. Drag the Widget until the position indicator is at the desired location, then release. In the figure below, List item 3 is being moved up to be between List item

1 and 2. The thin black outline (shown intersecting the labels of List item 1 and 2) moves as you drag the widget and the bold black line indicates where it will be placed.



4.3.7.3 Moving List Box Widgets

To move a List Box Widget:

- Click any List Box item to select the Widget.
A bold gray outline will appear around the entire List Box Widget.
- Grab the bold gray outline (not the middle of the Widget, as with simple widgets or when rearranging list items) and drag the entire List Box Widget to a new location in the Frame.

4.3.8 Pull-Down Lists: More Complex Behavior

 Pull-Down List Header Widget tool

Like the List-Box Widget, a standard Pull-Down List Widget is created as a set of items combined into one Pull-Down List that is automatically aligned and can move together as one Widget. In addition, CogTool knows how Pull-Down List items work together and produces that behavior for you automatically. It is *possible* to simulate this behavior using the custom Pull-Down List Header Widget and Pull-Down List Item Widgets (only available in custom mode), but it will involve many Frames and Widgets to do so.

The biggest advantages of using standard Pull-Down Widgets is realized when you use Display Labels.

Except for creating, changing the order of items, and moving, Pull-Down Widgets respond to all the common actions on Widgets, as described in a previous section.

4.3.8.1 Creating Pull-Down List Widgets

To create a Pull-Down List Widget, follow steps very similar to creating a List Box Widget:

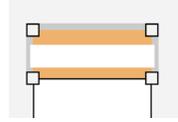
- Select the Pull-Down Header List tool  from the Widget toolbar.
- Drag a rectangle across an area of the Frame large enough to contain the longest

Display Label in the Pull-Down List.



Make the rectangle wide enough to contain the Pull-Down List item that has the longest Display Label.

- A Widget appear that has one highlighted Widget (in orange, below) with its Display Label ready to receive typing (white box). Type the Display Label for the first item and it will appear in the white box that appears on the Widget.



The first item of a Pull-Down List Widget is special. It acts as a header of the list rather than a member of the list itself. As soon as a member of the list is selected, this header will disappear from the user's view of the list. First items tend to be something like “[None]”, “Please select an item...”, or it can even be left blank. Type in whatever you want your user to see before picking any item from this pull-down list.

- Pressing the Enter key will create the first list item and move the cursor to the list item below the first.



- Continue to label the items by typing in the white box and hitting Enter to get to the next item, until all the items in that part of the Pull-Down List.
- The Pull-Down List Widget will contract to a single item, the first (header) item, when you stop typing and click anywhere else in the Frame window, e.g., on the Widget properties pane.
- You can determine which list item appears when the Pull-Down List Widget is contracted by choosing that item from the **Initially Selected Item** pull-down list at the bottom of the Widget's property pane (See Figure 4-25). If you leave that pull-down at **None Selected**, the special first item (the header) will be displayed when entering this Frame.



The first item of a Pull-Down List Widget acts as a header and will disappear from the displayed Widget if another item is selected.

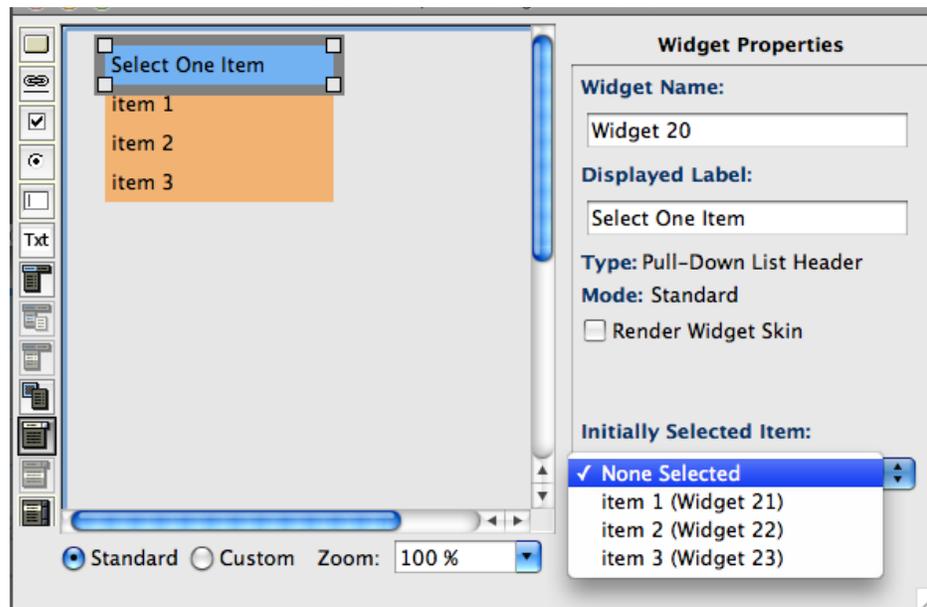


Figure 4-25 : The Frame window with a Pull-Down List Widget. If items 1, 2 or 3 were chosen as the Initially Selected item (from the pull-down list at the bottom right), “Select One Item” would not be displayed in the prototype.

4.3.8.2 Rearranging Pull-Down List Items

To change the order of a Pull-Down List item:

- Click the displayed Pull-Down List item to expand the Pull-Down List.
- Drag the desired Pull-Down List item to a new location within the same Pull-Down List Widget or to another Pull-Down List Widget.



When you drag a Pull-Down List item, an outline of the item and a bold black position indicator will appear. Drag the item until the position indicator is at the desired location, then release. (See Figure 4-26).



The first item of a Pull-Down List Widget acts as a header. Other list items cannot be dragged on top of the header. If you try to drag an item above the header, the cursor will change to a slashed circle to indicate that you cannot do that action. If you want to change the header, type in a new word in the header Display Label.

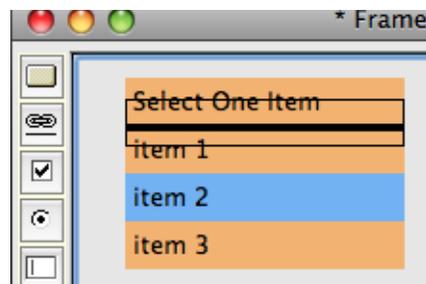


Figure 4-26: Pull-down list item 2 is selected for reordering, drag it until the bold black line is in the position you want, then release.

4.3.8.3 Moving Pull-Down List Widgets

To move a Pull-Down List Widget:

- Click the displayed Pull-Down List item to select the Widget.
A bold gray outline will appear around the entire Pull-Down List Widget.
- Grab the bold gray outline (not the middle of the Widget, as with simple widgets or when rearranging list items) and drag the Pull-Down List Widget to a new location in the Frame.

4.3.9 Menus and Context Menus: Highly Complex Behavior

The complex behavior of a hierarchical menu system, be it with regular menus at the top of a screen or window, or context menus that pop-up when you right-click on an object, is automatically prototyped for you using standard Menu and Context Menu Widgets. Like List-Box and Pull-Down List Widgets, a standard Menu or Context Menu Widget is created as a set of items combined into one. The items are automatically aligned and can move together as one Widget. In addition, CogTool produces the behavior of opening and closing menus for you automatically. It is *possible* to simulate this behavior with a combination of custom Menu Header, Submenu, Menu Item and Context Menu Widgets, but it will involve a multitude of Frames and Widgets to do so as completely as standard Menus and Context Menus do.

The biggest advantages of using standard Menu or Context Menu Widgets is realized when you use Display Labels.

With the exception of creation, operations on a menu system are the same for both the type of menu system that is always visible (e.g., at the top of a screen (Mac) or a window (Windows)) and context menus that pop-up at the cursor when invoked from an object on the screen (usually with right-click on Windows or CTRL-click on Mac). Operations that are the same will be described only once using examples from menu systems that are always visible.

4.3.9.1 Creating a Menu System

When creating menu Widgets in standard mode, CogTool makes creating an entire menu system a simple matter of typing in the menu headers, submenus, and menu items. You can create menus that are always visible, like the ones at the top of the screen in a Mac or at the top of windows in Windows, or context menus that pop-up at your cursor.

4.3.9.1.1 Creating Menus that are Always Visible



Menu Header Widget tool

To create a menu system that is always visible:

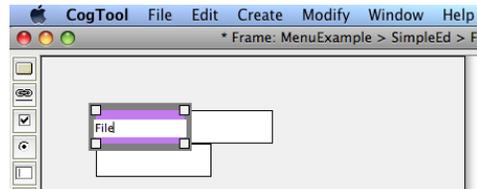
- Select the Menu Header tool  from the Widget toolbar.
- Drag a rectangle across an area of the open Frame. Try to size this first menu

header Widget to fit the longest label in the set.

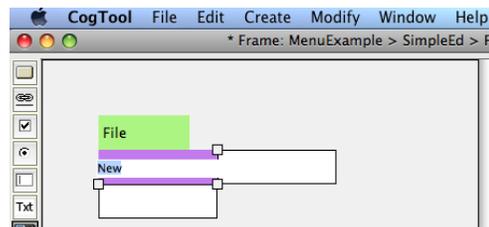


Although you can re-size the entire menu system later, it will be easier if you make the rectangle wide enough to contain the Menu Header that has the longest Display Label.

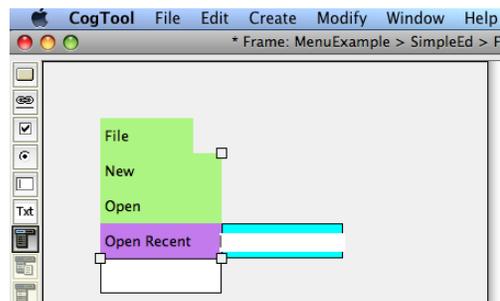
- Type you fist menu header into the white box that appears in the retangle.



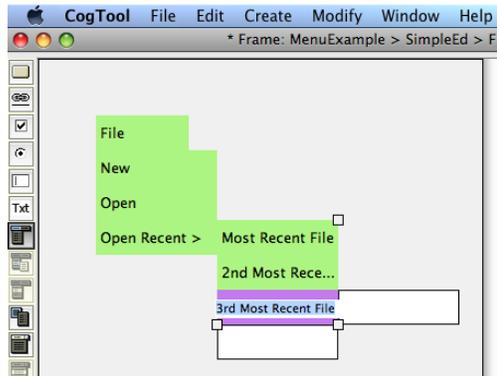
- Hit Enter, This creates the File Menu Header and puts your cursor into the white box of the first menu item. Type the first menu item.



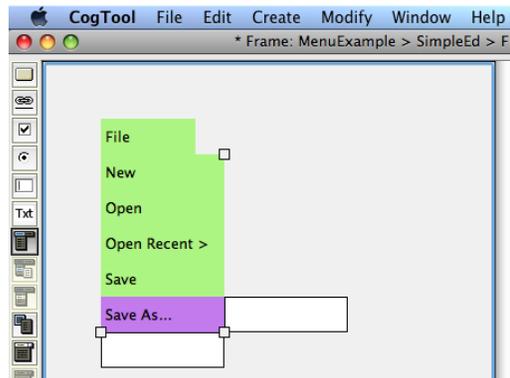
- Each time you hit Enter, CogTool creates a new menu item (the one you just typed in) and a spot to enter the next menu item. So just keep typing to enter in as many menu items as your system requires.
- To make a submenu with items that expand out to the side, select any menu item and click in the white box to its right, or use the arrow keys to navigate to the menu item and out into the white box on the right.



- As before, start typing in the top white box and every time you hit Enter, CogTool creates a submenu item and new spot for another one. When a submenu is created, CogTool automatically puts an arrow pointing to the right, to indicate that a submenu is available at this menu item.



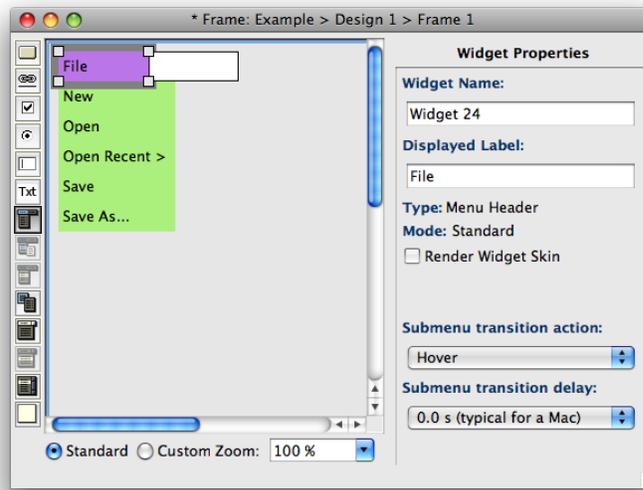
- At any time, you can navigate around this menu system either by clicking on the menu item you want or navigating to it with the arrow keys.. You can then add more items as necessary. If you navigate off of a submenu, it will contract (as below), but its arrow pointing to the right indicates that its submenu will reappear if you select that item again.



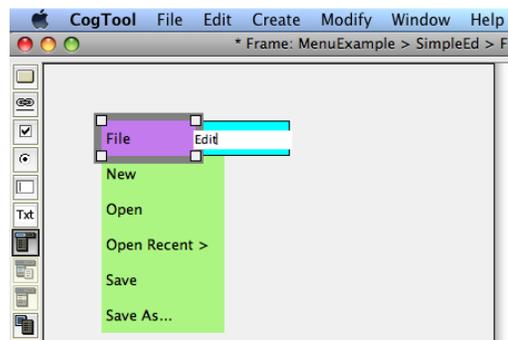
- Navigate up to the menu header (File) and examine the menu properties. There is a unique Widget Name automatically assigned by CogTool (in this case, “Widget 2”). You can change it if you wish, but it is not necessary for the proper operation of CogTool. The Display Label is the words you typed in (in this case, “File”). The center of the properties pane is the same as for any other Widget. Below Capture Background, there are a few items specific to Menu Headers. If you have submenus under this header, you can set whether they will open by a click of by hovering on them. Hover is chosen by default and if leave it, then you can set the time that the system waits before opening the submenu to 0.0 seconds (typical for a Mac) or 0.5 seconds (typical for a PC).



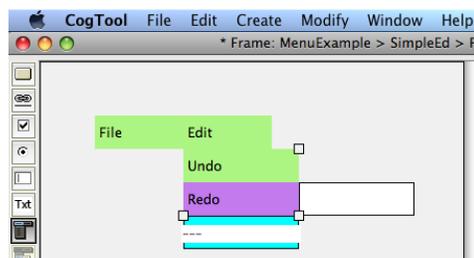
Each menu header has Hover and 0.0 seconds chosen by default. If you do not want the default, you must change each Menu Header separately in its property pane.



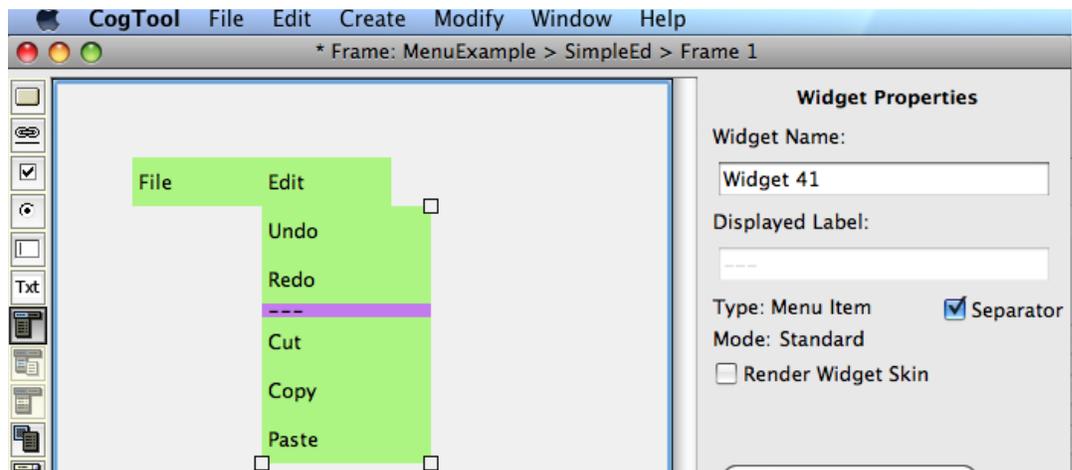
- To add more menu headers, navigate back to the top row, and a spot for a new header appears to the right. Type the menu header name into it and hit Enter, then continue making its menu items as before.



- To make a menu separator, type “---” as the menu item.



- When you hit Enter, the --- will be turned into a special menu item called a Separator, which is indicated by the checkbox under the Display Label in Widget properties pane. A Separator cannot be selected when demonstrating Tasks on a menu system. You can always change a Separator back to a regular menu item by un-checking the checkbox.



4.3.9.1.2 Creating a Context Menus



Context Menu Widget tool

To create a context menu and its children (submenus and menu items):

- Select the Context Menu tool  from the Widget toolbar.
- Drag a rectangle across an area of the open Frame where a user would click to bring up the context menu.



Make the rectangle big enough to contain the item that will be clicked on to pop-up the context menu. If it is a word, then you can use a Display Label in this widget. If it is a picture, you can use a background image (either on the Frame or in this Widget).

- Check the properties pane to make sure the values are as you wish. If there are submenus the action and delay are set as with the menus that are always visible. The action that brings up the context menu is set at the very bottom (right click (PC) or CTRL-click (Mac)).
- Type the labels in for the context menu items, hitting Enter between each one. Do not worry if the menu item labels do not fit in the displayed box at the moment, you can change the size of all of them at once after typing them all in. Add submenus as needed to prototype your design.

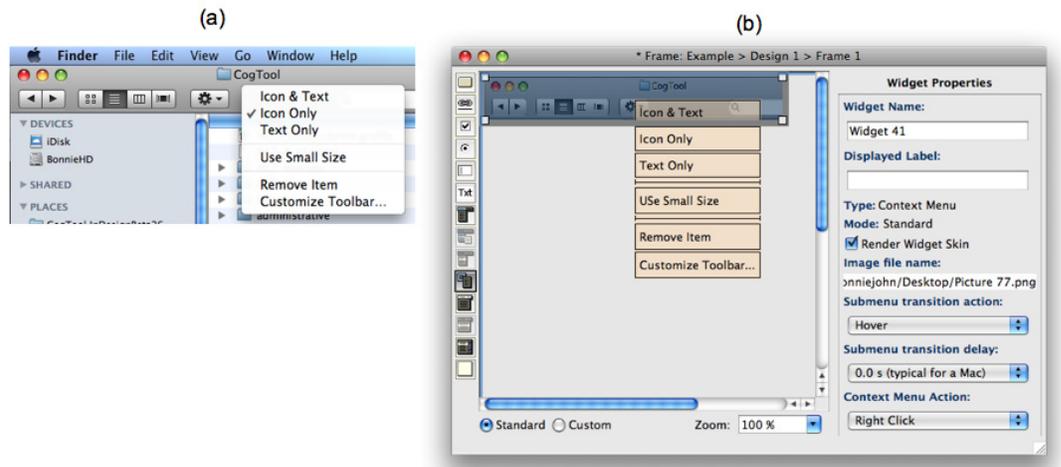
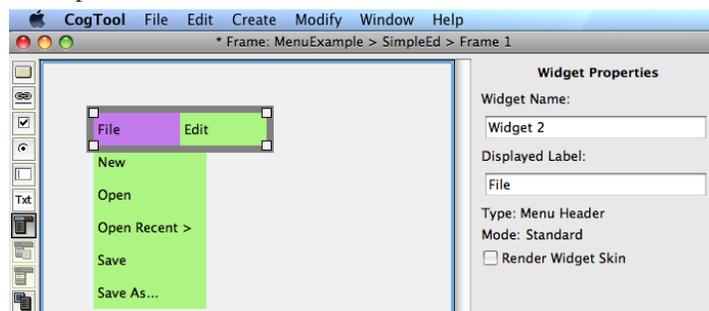


Figure 4-27: The right side (a) shows the behavior we would like to prototype, a context menu that comes up if you ctrl-click anywhere on the Firefox toolbar the Mac. The left side (b) shows how it would be done in CogTool. Create a Context Menu Widget with an image of the Firefox toolbar as its image. Type in the items in the menu, with separators, into the fields in the widget. Set the delay time and the action in the properties pane.

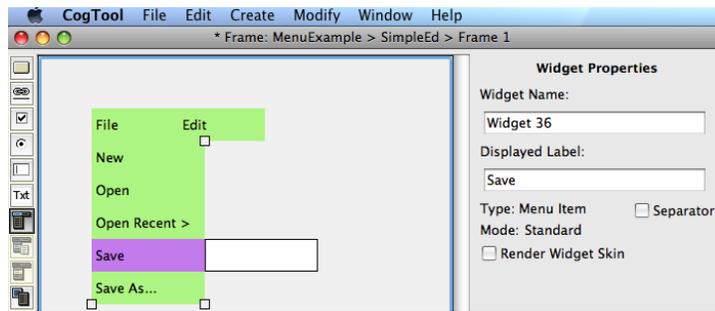
4.3.9.2 Selecting Menu Widgets

Menus are complex Widgets with many parts, consequently, selection is more complex than with simpler Widgets. The examples are from menus that are always visible, but selection works the same for context menus.

- To select a single item in a menu system, click on it. It will highlight by turning a different color from the Widget layer color (in this section, the Widget layer color is green and purple indicates selection). The Widget's properties are displayed in the properties pane.



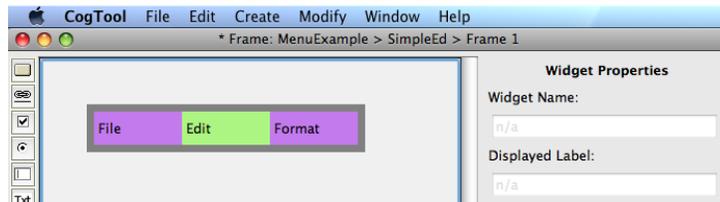
- The File menu header widget is selected in the picture above. The thick gray outline around the menu headers when a Menu Header is selected (e.g., File), indicate where to grab to drag the whole menu system around to place it where you want it in the Frame (see Moving Menu Systems). The little white boxes in the corners indicate where to grab to re-size the menu headers. (See Re-Sizing Menus).



- Save is selected in the picture above. The little white boxes in the corners indicate where to grab to re-size the menu items. (See Re-Sizing Menus)

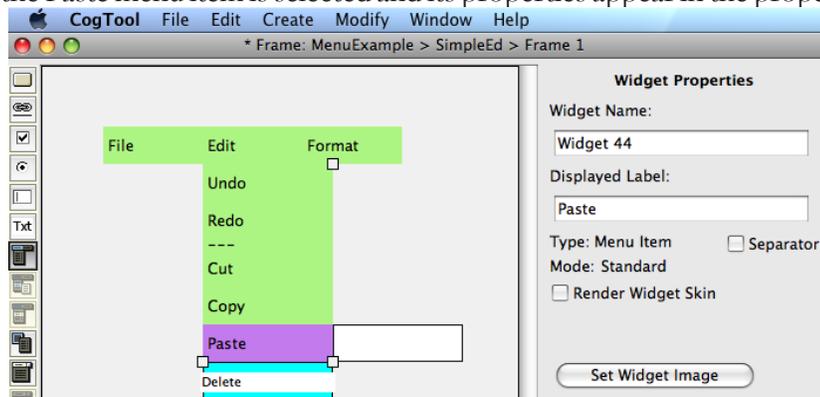
To select multiple menu headers

- Hold down the Shift key and click on all the items you want to select. Each one will turn to the highlighting color. File and Format are selected, below. When multiple menu headers are selected, the Widget properties pane does not show the properties of any Widget.



You cannot select multiple menu items, only menu headers. Menu items must be manipulated one at a time.

When you are creating menu items, the item you are in the middle of creating is not selected until you hit Enter (because it doesn't exist as a menu item until you hit Enter). Thus, while you are typing in the name of a menu item, the menu item above it is selected, is highlighted, and its information is in the properties pane. In the picture below, "Delete" has just been typed in, which is indicated by the white box surrounded by light blue. But the Paste menu item is selected and its properties appear in the properties pane.



4.3.9.3 Re-sizing Menu Widgets

To change the size of menu widgets,

- Select a menu header or item. That item will highlight in a color opposite to the Widget layer color, and small white squares will appear at the corners of the column (for submenus or menu items) or row (for menu headers).
- Position your cursor on one of the white squares and drag it to the desired size.

4.3.9.4 Moving Menu Widgets

To change the location of the entire menu system in a Frame,

- Select one of the menu headers. That menu header will be highlighted and a thick gray outline will appear around all the menu headers.
- Position your cursor on the thick gray outline and drag it to the desired location.

To move a Widget (menu header, submenu, or menu item) within a menu system:

- Click the Widget to select it. It will highlight in a color different from the Widget color (e.g., the Widget layer is green and the highlight is purple).
- Drag the Widget to a new location within the same menu system. When you drag a Widget, a halo, or outline, of the Widget and a black position indicator will appear. Drag the Widget until the position indicator is in the location you wish to move the Widget, then release.

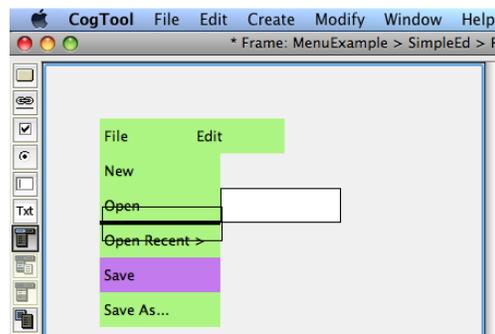


Figure 4-28: The Save menu item in being moved up in the menu. Save is purple because it is the menu item being moved. The black outline moves with your cursor as you drag the menu item around. The black bar indicates that if the mouse button were to be released now, the Save menu item would appear in between Open and Open Recent.

FYI Holding the Option key (Mac) or Control key (Windows) while dragging duplicates the menu item, leaving the original in place.

FYI You can move a menu header and all its items will move with it. Likewise, you can move a submenu and all its items will move with it.



You can move a menu item within its menu (as shown above), or another menu or submenu in the same menu system, or even to another menu system in the Frame. When moving menu items to menus other than its own, hover over the target menu long enough to let it expand so you can properly place the item you are dragging. Do not release the mouse button until the black bar is in the position you want.

4.3.9.5 Copying/Cutting/Pasting Menu Headers and Their Items

When you copy a menu header within a menu system, you are also copying all the submenu and menu items that are under the menu header.

To copy a menu header:

Mac: ⌘C
Win: CTRL+C

- Click the menu header to select it.
- From the **Edit** menu, choose **Copy Widget**.

To cut a menu header:

Mac: ⌘X
Win: CTRL+X

- Click the menu header to select it.
- From the **Edit** menu, choose **Cut Widget**.

To paste a menu header:

Mac: ⌘V
Win: CTRL+V

- Click the menu header to select it.
- From the **Edit** menu, choose **Paste**.



Although you can select individual submenu or menu item Widgets, cannot copy, cut, and paste them. When one of these items is selected, the Cut and Copy items in the CogTool Edit menu are grayed out and the keyboard shortcuts do not work.

4.3.9.6 Copying/Cutting/Pasting an Entire Menu System

To copy an entire menu system:

- Select all the menu headers in the system using Shift-Click, or by holding Shift while dragging a rectangle across them. Be sure to capture all the menu headers you want to copy..

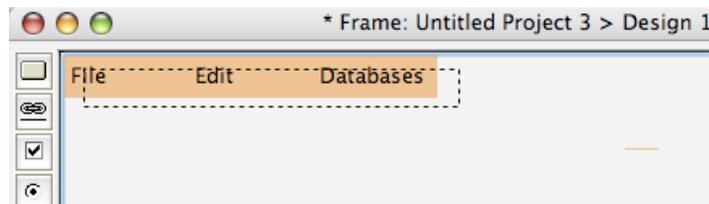


Figure 4-29: Selecting all the menu Widgets in a menu system by holding Shift while dragging across them.

Mac: ⌘C
Win: CTRL+C

- From the **Edit** menu, choose **Copy Widgets**.



When you copy all the menu headers in a system, you are also copying any submenu and menu item Widgets connected to them (thus, the entire menu system).

To cut an entire menu system:

- Select all the menu headers in the system by holding Shift while dragging a rectangle across them.
- From the **Edit** menu, choose **Cut Widgets**.

Mac: ⌘X

Win: CTRL+X

To paste an entire menu system:

- Select all the menu headers in the system by holding Shift while dragging a rectangle across them.
- From the **Edit** menu, choose **Paste**.

Mac: ⌘V

Win: CTRL+V



If you select non-contiguous menu headers when you copy or cut, the gaps between them will be removed when you paste

4.3.10 Working with Text Boxes & Text



Text Box Widget tool



Text Widget tool

Text Box Widgets and Text Widgets are used together to prototype interactions with editable text in an interface. Text box Widgets typically contain Text Widgets, just as text boxes in an actual interface contain editable text.

In a real interface, user do three types of things in a text box: (1) click on it to place the cursor there so they can type (2) type into an empty text box, and (3) edit text that is already in a text box.. If your prototype will only need to mock-up the first two actions, clicking and typing into an empty box, then a Text Box Widget is all you need. If, however, you need to prototype more sophisticated text interactions (e.g. dragging over some text to highlight it, clicking before words to add new text, clicking at the end of text to insert more, etc.) then you need to use text Widgets as well.



If your prototype needs only clicking in an empty text box and entering text, then you can use a Text Box Widget alone.



If your prototype needs to select or edit existing text, you must use Text Widgets, too, usually inside a Text Box Widget.

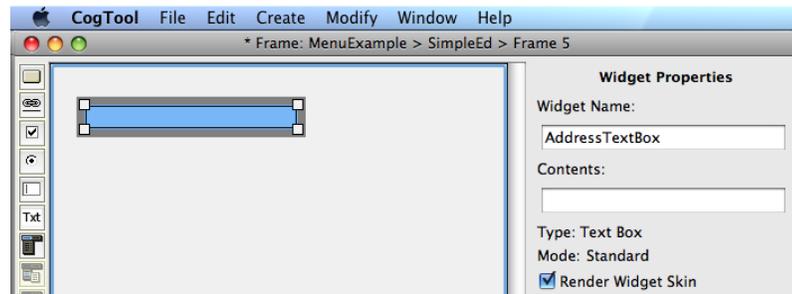
All common operations on simple Widgets apply to Text and Text Box Widgets.

4.3.10.1 Creating a Text Box

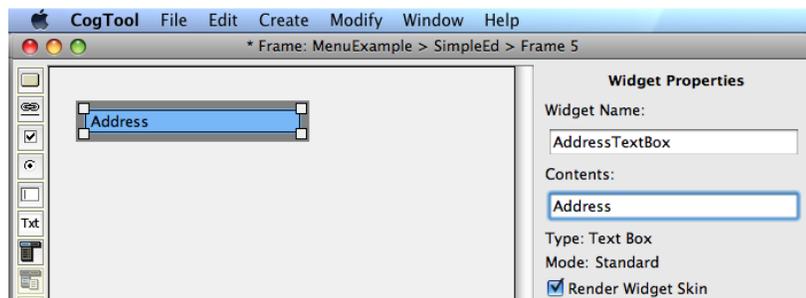
 Text Box Widget tool

To create a text box Widget:

- Select the text box tool  from the Widget toolbar.
- Drag a rectangle across an area of the Frame where you want the text box to be.



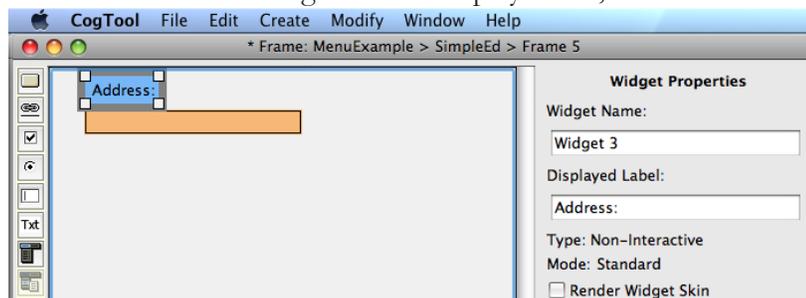
- If you want this text box to be empty when this Frame is entered, then type in a meaningful Widget name for this text box (e.g., as above, "AddressTextBox"). Do not type anything into the Display Label, because this will be displayed inside the text Box Widget, as below.



If you want your text box to have content when the prototype enters this frame, then put that content in the Display Label. For example, if you are prototyping a web site that remembers people's address after they have entered it, the first frame with an address text box will not have a Display Label in the Text Box Widget, but the second Frame will.



If you want your text box to have a label above it in the Frame, use a Non-Interactive Widget with a Display Label, as shown below.





Text boxes Widgets can be any size, and their display labels can be any number of characters, but CogTool will only show one line of text. If your interface requires multiple lines of text, create draw a Text Box of the correct size and use multiple Text Widgets inside it to simulate multiple lines of text. Or, if you are not editing the text, use a background image showing the multiple lines of text in your interface.

4.3.10.2 Creating Editable Text, an Example

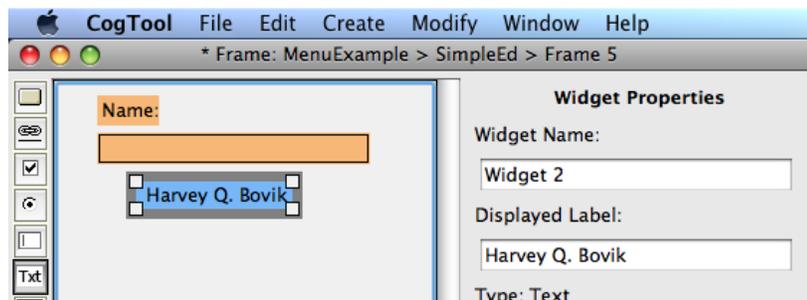
Consider the task of fixing an error in the name entered in a text box. The name is supposed to be “Harry Q. Bovik”, but someone entered “Harvey Q. Bovik”. There are several ways of fixing this text, and the following examples will demonstrate how to use Text and text Box Widgets to prototype three of them.

Most prototypes of editable text start with creating a Text Box Widget and a Text Widget with the text-to-be-edited inside it.

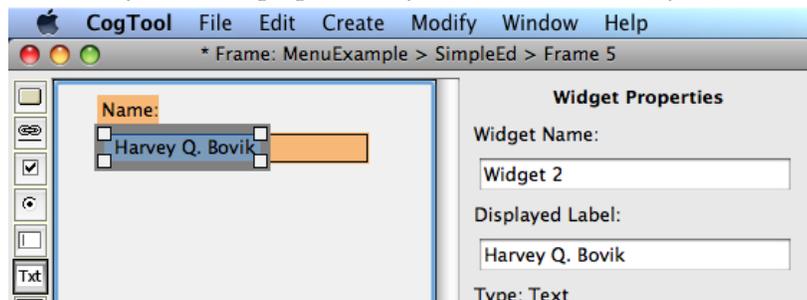
- Create a Text Box Widget as shown in the previous section.
- Select the text tool  from the Widget toolbar.
- Drag a rectangle the size of the text you are going to edit. This is typically smaller than the Text Box Widget it will be inside. The type in the text you want to edit into the Display Label.



It is easier to work with the Text Widget if you create it outside the Text Box Widget (se shown below) and move it inside afterwards.



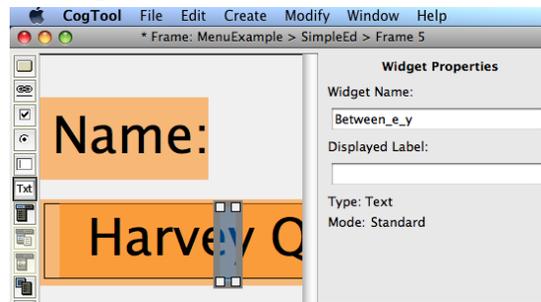
- Move the Text Widget into the Text Box Widget. You are now ready to prototype several ways of changing “Harvey Q. Bovik” to “Harry Q. Bovik”.



4.3.10.2.1 Place the Cursor, Delete and Type the Right Letters.

In a real interface, a user might place the cursor between the “e” and the “y” in “Harvey”, and then delete the “ve” and type “r”. To set up the Frame so this method can be prototyped, there needs to be a target widget between the “e” and the “y” to click on.

- Create a tiny Text Widget outside the Text Box Widget. Give it a meaningful name (e.g., “Between_e_y”)
- Move it to between the “e” and the “y”. (The picture below is zoomed in to 400% to make this widget more visible.)

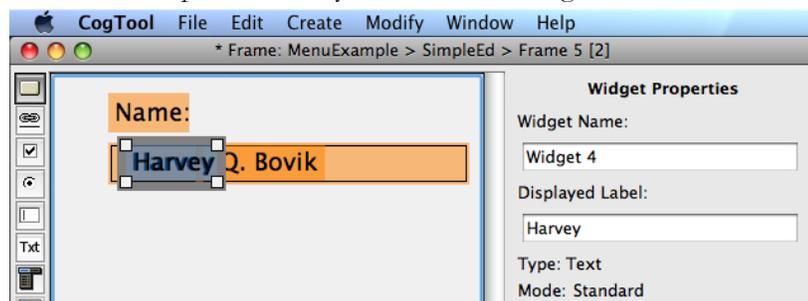


Now your Frame has a target for the click that is needed to start this method. We will see how this plays out when defining Transitions in the next section and demonstrating this task in the next Chapter.

4.3.10.2.2 Double-Click on the Incorrect Name, Type the Correct Name.

In a real interface, a user might double-click on “Harvey” to highlight it, then type “Harry”. To set up the Frame so this method can be prototyped, there needs to be a target Widget on “Harvey” to double-click on.

- Create a Text Widget the size of Harvey outside the Text Box Widget. You can either give it a meaningful name (e.g., “Harvey”) or let it have a Display Label of “Harvey”. (Either way, “Harvey will appear in the Frame.)
- Move it to lie on top of “Harvey” in the Text Widget.

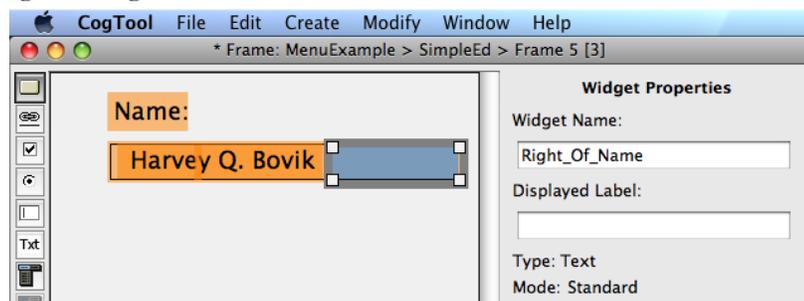


Now your Frame has a target for the double-click that is needed to start this method. We will see how this plays out when defining Transitions in the next section and demonstrating this task in the next Chapter.

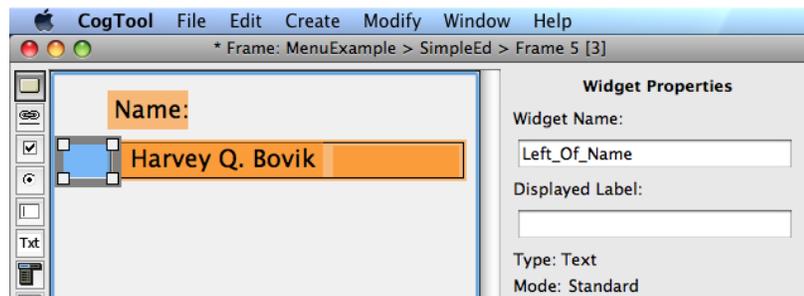
4.3.10.2.3 Drag to Highlight the Incorrect Name, Type the Correct Name.

Finally, a user might drag over the entire name highlight it, then type “Harry Q. Bovik”. To set up the Frame so this method can be prototyped, there needs to be a target Widget to press the mouse-button down on (starting the drag) and a target Widget to release the mouse-button on (ending the drag). If the user were to drag from right to left, the mouse would have to be pressed in the text box to the right of the name and released anywhere to the left of the name.

- Create a Text Widget the size of that portion of the text box to the right of the name that is blank. Draw it outside the Text Box Widget and give it a meaningful name (e.g., “Right_of_Name”).
- Move it to lie in the blank area in the Text Box Widget. This creates a target for starting the drag.



- Create a Text Widget the height of the Text Box Widget and the width of the area that it would legal to release the mouse button in you interface (usually outside of the text box). Give this a meaningful name (e.g., “Left_Of_Name”) and position in the correct place.



Now your Frame has a target for the start of the drag and one for the end of the drag that will highlight the incorrect name. We will see how this plays out when defining Transitions in the next section (see “Dragging Over Text to Highlight it”) and demonstrating this task in the next Chapter (see “Demonstrating the Three Text-Editing Tasks Designed in Previous Section”).

The interplay of Text Box Widgets and Text Widgets is complex only when you need to prototype editable text. Fortunately, many systems can be usefully prototyped without simulating editing text..



Text Widgets can be any size, and their display labels can be any number of characters, but CogTool will only show one line of

text. If your interface requires multiple lines of text, use multiple Text Widgets to simulate multiple lines of text.

4.4 Transitions

4.4.1 What is a Transition?

A Transition represents the user's action that causes the system to move from one Frame to the next. Frames only depict the static pictures of how the system presents itself to the user; Transitions are necessary to represent the interactivity of the system.

When a user takes action on a Widget using a mouse or touchscreen, e.g., click, double-click, etc., the Transition is drawn from the Widget, indicating that the action actually happens at that Widget. When a user types, the Transition is drawn from the keyboard (in rare cases, described below, a typing Transition can be drawn from a Widget while still representing a user typing on the keyboard). When the user speaks to a system with a microphone, the Transition is drawn from the microphone.

Transitions are created in the Design window and are represented by black arrows between a Widget or input Device (keyboard or microphone) and a Frame. A Widget or Device can have as many Transitions as you need to represent your Design, but each transition from a Widget must be unique (e.g., a left-click on a Button Widget cannot transition to two different Frames).

The toolbar on the far left of the Design window contains tools for drawing types of Transitions action types:

	= default Transition type
	= mouse action (e.g. clicks and hover)
	= keyboard
	= touchscreen (e.g. taps and hover)
	= Graffiti®
	= Microphone



Because each Widget type has its own default set to the most common Transition from that Widget type, and because Transition types can be changed from the Transition properties pane, most CogTool users find that the default tool (selected by default) is the only tool they need.

After describing the operations possible on Transitions, this section examples of how to mock-up some common user actions, like drag-and-drop and editing text.

4.4.2 Creating New Transitions

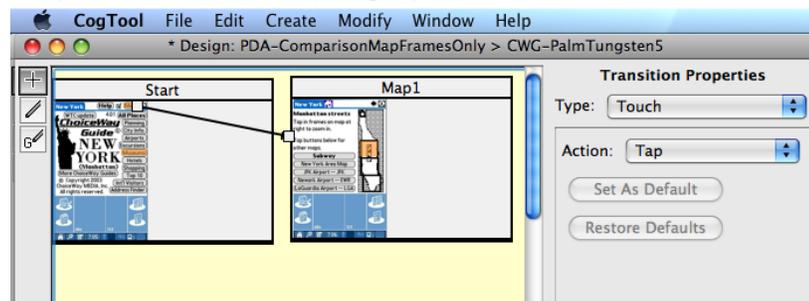
Transitions can emanate from a Widget that is operated on by a mouse or, when a touchscreen Device is included in the Design, by a stylus or finger. Transitions can also emanate from the keyboard or microphone, when these Devices are present in your Design. All Transitions are drawn in the Design window.

4.4.2.1 Transitions from a Simple Widget.

To create a Transition from a Widget to an existing Frame:

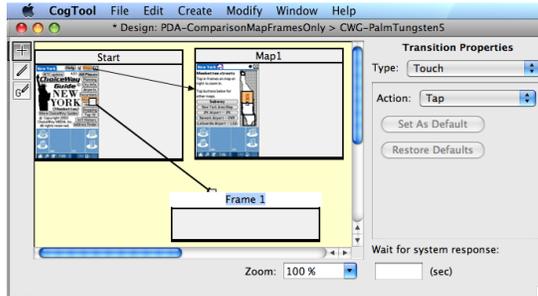
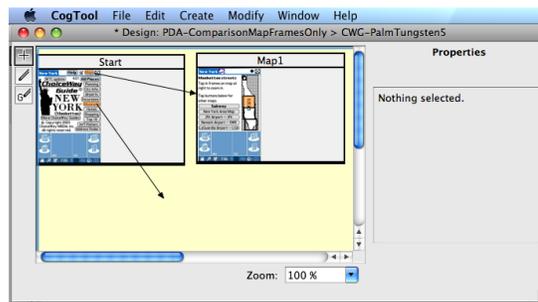
- Click on the Widget and drag the cursor to an existing Frame that is the result of an action on that Widget. Release the mouse button when the desired Frame highlights (turns gray). A black arrow will connect the Widget to the Frame, the Transition will remain selected (indicated by white squares at its ends) and the properties pane will contain the properties of the Transition.

For example, the picture below shows a transition drawn from a small Button Widget at the top of the “Start” Frame (the button is obscured by the Transition’s end) to the next Frame (“Map1”).



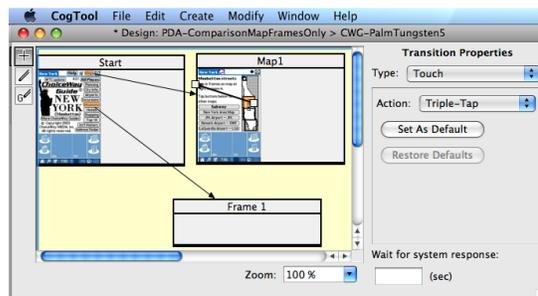
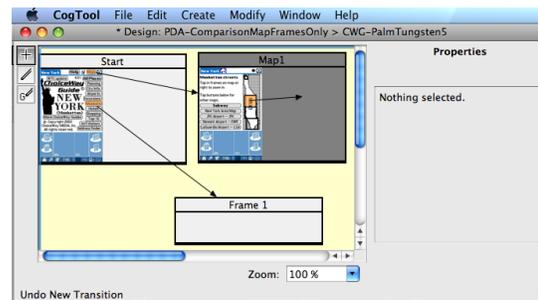
- Examine the Transition Properties pane to make sure it is the Transition you intended. Here, we are prototyping a Palm Tungsten-5, with a touchscreen, so the Type Touch is appropriate for a Button Widget. The Action is Tap (it could be double-tap, triple-tap, etc.). Other types of Transitions, e.g., using a mouse, have other options in their properties pane.

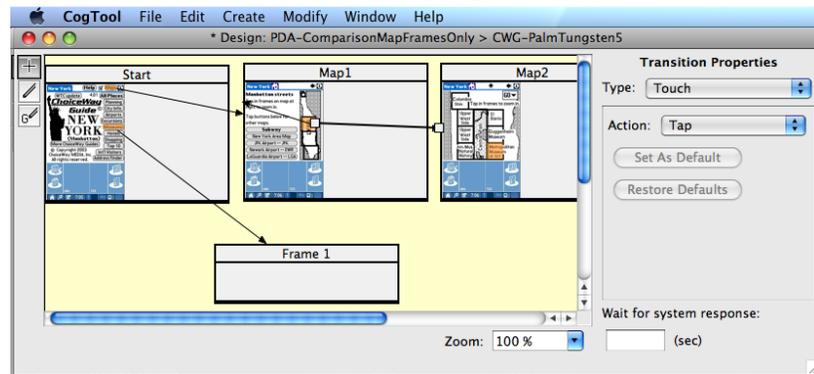
Transitions can also be drawn to a nonexistent Frame by dragging to an area of the Design canvas that is not occupied by a Frame, as shown below from a different Button Widget in the Start Frame. Doing so causes a new Frame to be created, which can then be populated with images and Widgets.



Transitions can also be drawn to the same Frame by dragging to an area of the same Frame outside the Widget that is the source of the Transition. The Frame will highlight and the Transition will snap to the nearest edge of the Frame when released. This is called a “self-Transition”.

Self-Transitions are used when you want to minimize the number of Frames you have to draw in a Design and the action represented by the Transition does not change the look of the system very much, or you don't want to draw intermediate Frames. For example, if a user had to tap the area on the map in Frame Map1 four times before the interface showed enough detail to continue the task, you could prototype this with four Frames each showing a more zoomed-in map with single-tap transitions between them or you could reduce the number of Frames and use a triple-tap self-transition and then a single-tap regular transition to the next Frame (shown below).



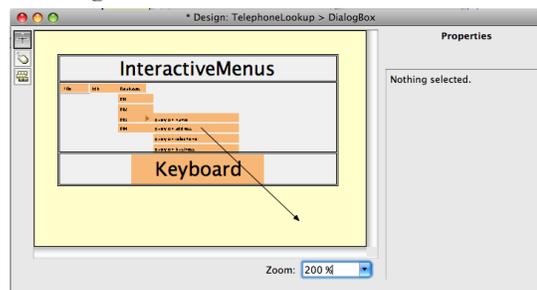


4.4.2.2 Transitions from an Interactive Widget: Standard Menus, Context Menus, and Pull-Down Lists

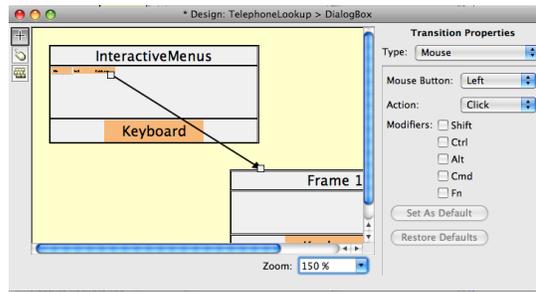
Transitions can be drawn from interactive widgets that change their appearance as you click on them in a Frame. Standard Menu Widgets, Context Menu Widgets and Pull-Down List Widgets open up when you click on them and the transitions are from the revealed items, not those that are visible without interaction.

To create a Transition from a complex Widget:

- Click on the interactive widget until the item you want to transition from is revealed.
- Drag from that item to a different exiting Frame, to an empty portion of the Design canvas to create a new Frame, or within the same Frame to create a self-Transition. The picture below shows an interactive menu opened to reveal a submenu and a Transition is drawn from the second item of that menu to an empty part of the Design canvas.



- When you release the mouse button on the destination Frame, the interactive Widget closes. There can be many Transitions from the component parts of an interactive Widget and they will all appear to emanate from the closed Widget. Click on the Widget to see the individual Transitions again.



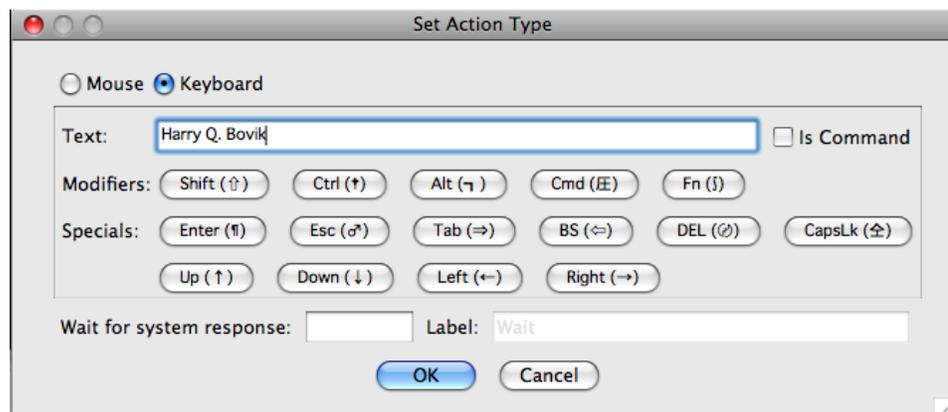
- As the above picture shows, mouse Transitions have additional options in the Transitions Properties pane. You can set which mouse button to use (left, middle or right), which action to perform (click, double-click, press, release, etc.) and which keys must be held down while performing the mouse action.

4.4.2.3 Transitions from the Keyboard or Microphone

When users type, they are performing actions on the keyboard, and thus, those actions are represented with Transitions from the keyboard. Likewise, when users speak to a system through a microphone, those actions are represented with Transitions from the microphone. The section “Working with Sound” details how to work with both the microphone and speaker, so examples here will be from the keyboard.

To create a Transition from the keyboard

- As with any Transition, click on the source of the user action, in this case, the keyboard Device at the bottom of the Frame, and drag the cursor to an existing Frame that is the result of typing on the keyboard. Release the mouse button when the desired Frame highlights. Self-Transitions and Transitions to a nonexistent new Frame are also allowed as keyboard Transitions.
- With a keyboard Transition, a dialog box appears for you to type in the characters the user will type to cause the system to change from one Frame to the next. Type the characters the user will type at this point in the prototype. Modifier or special characters are added by clicking their buttons. If you type upper-case letters, a shift is assumed, so clicking the Shift button followed by a “c” is the same as typing “C” into the Text field.

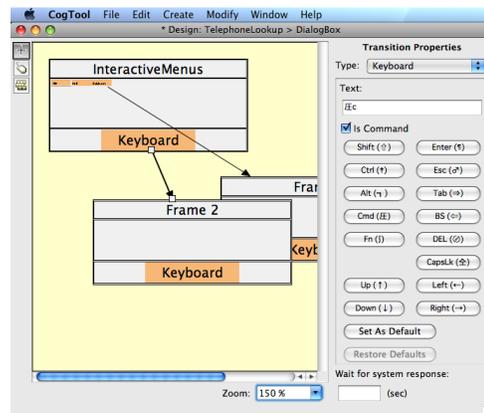


- *If the text the user types is not a command to the system*, that is, you are simply typing in text, like a name into a Name field, or an address into an Address field, then leave the Is Command checkbox in its default unchecked state.
- *If the text the user types contains special characters or modifiers*, use the buttons below the text field to enter the characters. CogTool will automatically treat these as commands when it creates its cognitive model and you can leave the Is Command checkbox in its default unchecked state. (Shift is available in the Modifier list for when it is needed in a command-key pattern, like Cmd-Shift-n to make a new folder in the MacOS. If you want a capital letter or symbol above the numbers on the keyboard, simply that character into the text field.)
- *If the text the user types is a command to the system but does not contain the special characters* (e.g., very old line editors like “ed” used regular characters as commands. For example, “\$p” was a command to print the last line of the file), CogTool has no way of knowing this is a command. Therefore, such an “unknowable command” must appear on a transition by itself and you must explicitly indicate its status by checking the Is Command checkbox.
- As long as there are no “unknowable commands” as described immediately above, feel free to put an entire string of text, regular characters, modifiers and specials into the text field. CogTool will parse them correctly and create a plausible cognitive model automatically.



Although it looks like you can change a keyboard Transition to a mouse Transition at the top of this dialog box, that is not the case. You cannot have a mouse Transition from the keyboard and the dialog box will gray out if you select the mouse radio button. (The CogTool team will fix this UI to make more sense soon.)

- After clicking OK, a black arrow will connect the Widget to the Frame, the Transition will remain selected (indicated by white squares at its ends) and the properties pane will contain the properties of the Transition.



Keyboard transitions that involve many special characters in a row result in predictions of human performance that are slower than how skilled computer users actually perform. This is because (1)

KLM assumes all keystrokes are done in sequence even for keys held down together and (2) ACT-R (the cognitive architecture making the predictions “under the hood”) provides a model of typing that is too simple, assuming each finger has to move back to the home-row before typing the next key. For most comparisons between interfaces, these assumptions are reasonable approximations to user behavior. However, for extremely skilled users of multiple-key keyboard shortcuts and navigation keys, or for tasks that are extremely short (e.g., “Save the file” is accomplished with a single keyboard shortcut command), the predictions will be too slow.



It is possible to create typing and voice transitions from a Widget instead of from the Keyboard or Microphone Devices. This is not recommended (because the user is not actually typing on the button on the screen or speaking into the text box) and can be avoided in all but the most unusual cases. Under rare circumstances, you might need to express two typing (or voice) transitions from the same Frame that have the same content but go to different destination Frames. CogTool does not allow such an ambiguous representation, but you can work-around this restriction by putting one of the transitions on the keyboard (or microphone) and one on a Widget in the Frame.

4.4.2.4 Transitions from a Graffiti® Widget

Transitions from a Graffiti® Widget are similar to those from the keyboard in that a dialog box appears that allows you to type in the Graffiti® gestures the user will input and specify whether those gestures are a command to the system, or data being entered into a field. In this case, CogTool does not recognize special characters, so all commands must be on separate transitions and explicitly indicated with the Is Command checkbox.



The only characters CogTool will accept for Graffiti® gestures are letters, digits, comma, period, semi-colon, single-quote, dash, forward slash, back-slash, equal sign, back-quote and left and right square brackets.

4.4.3 Prototyping a System Delay

Just as each user action can cause a system delay, each Transition includes a place to specify how long a delay to insert. At the bottom of each Transition properties pane, is a text box labeled “Wait for system response” and is measured in seconds (sec). The system will wait for that number of second after completing the user action and before presenting the next Frame.



Numbers can only be entered into CogTool in US format, i.e., using digits and a “.” to represent the decimal point.

If you enter a number into this text box, you may then label the system delay if desired. For example, you may want to label a delay “Network lag”, “Searching”, or “Speech recognition”, to help you remember what the system is doing during significant delays.

If you are using CogTool to benchmark performance against an existing system, you are likely to use system delays so that the time for the existing system is accurately recorded. If you are comparing design ideas for a new system, you may not want to include system delays, but compare those ideas under the best possible conditions (i.e., no perceptible delay).

4.4.4 Modifying a Transition

4.4.4.1 Changing the Properties of a Transition

Transitions have properties that are initially set when the Transition is created and are specific to each type of Transition.

These properties can be viewed and changed at any time in the panel to the right of the Design canvas.

Mac: ⌘E

Win: CTRL+E

You can also edit a Transition by selecting it, then, from the **Edit** menu, choose **Edit Transition**. A dialog box appears with the same options for editing the Transition as are present in the properties pane.

4.4.4.2 Changing the Source and Destination of a Transition

You can change the source of a Transition to any other source on that or any other Frame that supports the action type of the Transition. Likewise, you can change the destination of a Transition to any Frame.

To change the source of a Transition:

- Select the Transition.
- Drag the white square located at the current source to the new source.



The mouse cursor will change to a crosshair when it is pointing at a Widget that can accept the Transition.

To change the destination of a Transition:

- Select the Transition.
- Drag the white square located at the current destination Frame to the new destination Frame.

4.4.5 Deleting a Transition

To delete a Transition:

- Select the Transition.



The Transition is selected when a white square appears at the endpoints of the Transition arrow.

Then do one of the following:

- Press the keyboard Delete key.

OR

- From the **Edit** menu, choose **Delete**.
- In either case, a dialog box will ask you to confirm the deletion.

4.4.6 Prototyping Some Common User Actions

Some user actions require more than selecting the correct action type from the Transition property pane. Several common instances are presented below.

4.4.6.1 A Series of Clicks/Taps on the Same Button (e.g., Buttons that Navigate or Scroll)

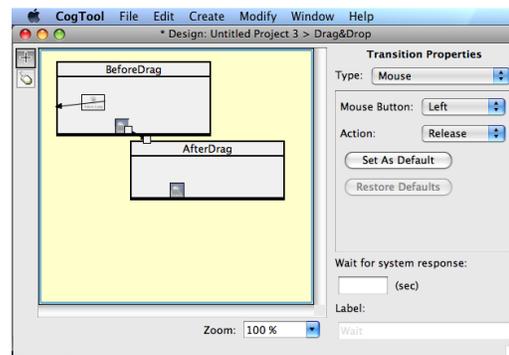
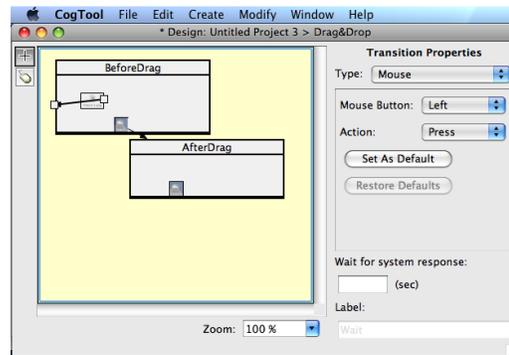
Sometimes a user will need to click multiple times on the same button, most commonly when they are scrolling or navigating with arrow buttons (e.g., changing the month in a date-picker). In this case, the same type of user action on the same button does different things, i.e., the first few clicks make progress but the last click attains the desired state.

You could choose to make the result of each click a separate Frame with the progress depicted and then each Frame has its own button with one click to take it to the next Frame. However, this does not convey any additional information about how the interface works, e.g., everyone knows that each click on the Next-Month arrow of a date-picker will advance the calendar one month, and it does take time to create these Frames and increases the size of the CogTool file. So you may decide to use self-Transitions to represent the first few clicks with only the last click Transition going to the Frame with the desired result. However, each Widget in a Frame can only have unique Transitions emanating from it, so each button Widget can only have 1 single-click, 1 double-click and 1 triple-click Transition. Use a combination of these Transitions to simulate everything from one to six clicks on the same button before transitioning to a new Frame. For example, a self-Transition triple-click plus a regular Transition single-click represents four clicks on the same button. If your design requires more than six clicks on the same button, you will need to put in one or more intermediate Frames to prototype the interaction.

4.4.6.2 Drag-and-drop

Drag-and-drop is represented in CogTool as a “press” on the mouse button at the start of the drag and a “release” of the mouse button at the end of the drag. Thus, you must have a Widget at the start with a press Transition and a Widget at the end with a release Transition. The following pictures show the representation of dragging a file to the trash and dragging the elevator in a scroll bar half-way down the bar.

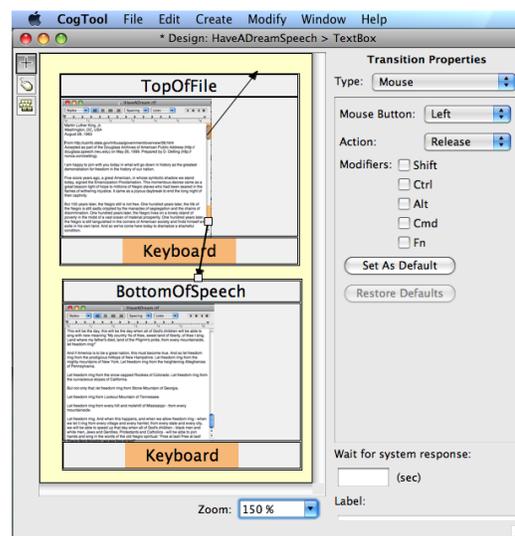
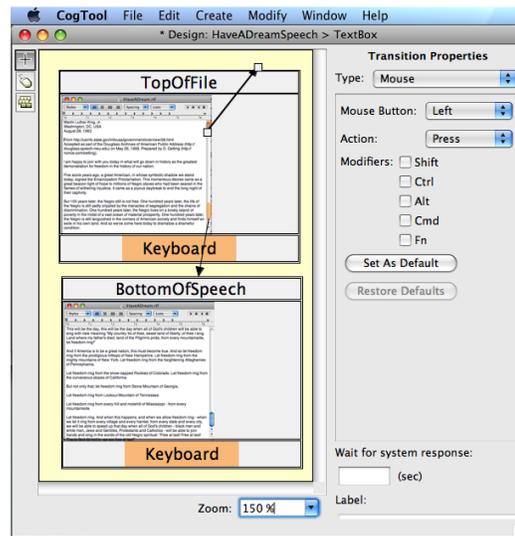
To simulate drag-and-drop a file to the trash, put two button Widgets in a Frame, one for the file and one for the trash. Make a second Frame with just the trash Widget. Draw a self-Transition from the file Widget with action type “press” This represents a user pressing the mouse button down on the file icon (top picture). Draw a Transition from the trash Widget with action type “release” to the second Frame. This represent that when the user releases the mouse button on the trash icon, the file is now in the trash and it is no longer visible on the screen (bottom picture).



CogTool does not enforce that “press” and “release” come in pairs. When demonstrating a task using drag-and-drop, you must be careful to demonstrate first the “press” action and then the “release” action without any other mouse actions in between.

To simulate moving the scroll elevator from the top to the bottom of the file, have two Frames, one with the contents of the window as it would appear before the scroll and the other as it would appear after the scroll. Put two button Widgets in first Frame; one on the elevator shown at the top of the window and one where the elevator will end up at the bottom of the window. Draw a self-Transition from the top Widget with action type “press”. This represents a user pressing the mouse button down on the scrollbar elevator (top picture). Draw a Transition from the Widget at the bottom of the win-

down with action type “release” to the second Frame. This represent that when the user releases the mouse button at the bottom of the scrollbar, the window has scrolled and the bottom of the file is now visible (bottom picture).



CogTool does not enforce that “press” and “release” come in pairs. When demonstrating scrolling in a window, you must be careful to demonstrate first the “press” action and then the “release” action without any other mouse actions in between.

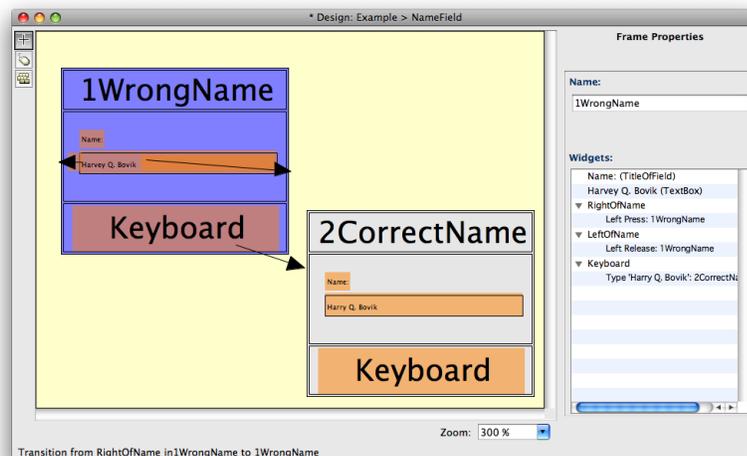
4.4.6.3 Dragging Over Text to Highlight it

In the section on Working with Text Boxes and Text, we showed how to place text Widgets in preparation for demonstrating how to simulate editing text. Here we show how to draw Transitions in the example that drags over the incorrect text in order to highlight it.

The Frame that contains the incorrect text has two text Widgets, one to the right of the incorrect name and one to the left of the incorrect name. To simulate dragging across

the text from right to left, draw a self-Transition from the Widget to the right of the incorrect name with the “press” action. Draw another self-Transition from the Widget to the left of the incorrect name with the “release” action. One additional Transition is needed to complete this Design: a keyboard Transition that contains the letters of the correct name (Harry Q. Bovik) to a new Frame representing the corrected text.

The figure below shows the prototype of this interaction, two Frames in the Design Window and the properties of the first Frame (“1WrongName”). This Frame contains five Widgets (TitleOfField whose display label is “Name:”, TextBox whose display label is “Harvey Q. Bovik”, RightOfName with no display label, LeftOfName with no display label, and Keyboard.). There are three transitions (self-transition “press” on RightOfName, self-transition “release” in LeftOfName, and a typing transition to the other Frame “2CorrectName” with letters “Harry Q. Bovik”)



CogTool does not enforce that “press” and “release” come in pairs. When you are simulating a drag user action, you must be careful to demonstrate first the “press” action and then the “release” action without any other mouse actions in between.

4.5 Working with Sound

There will be times when you will want to have sound capabilities with your prototype. You will either want to have the user issue voice commands or talk to another person or have the system generate informational or alert sounds or have other people talk to the user (e.g. the Air Traffic Controller giving instructions to a pilot). CogTool has devices for both microphone and speaker.



CogTool does NOT actually play or record sound. It represents using sound in a interface using text.

4.5.1 Microphone

The microphone represents voice input from a user. You must have a microphone Device included in your Design. When you draw a Transition from the microphone box at the bottom of the Frame, a dialog box appears and asks for the words the user will speak. You can also enter these words by selecting the Transition and entering them in the Transitions Properties pane (see Figure 4-30).



If you want valid quantitative predictions of task execution time, you must write out the words people have to say. For example, enter “one two three” instead of “1 2 3.”

The “Is Command” property functions the same way for microphone Transitions as it does for keyboard Transitions; it distinguishes whether spoken words should be recognized as a command, (e.g. when you tell your cell phone to “Call Mom”) which triggers an active behavior in the modeled system, or as simple voice input (e.g. when the pilot acknowledges the Air Traffic controller’s instructions.)

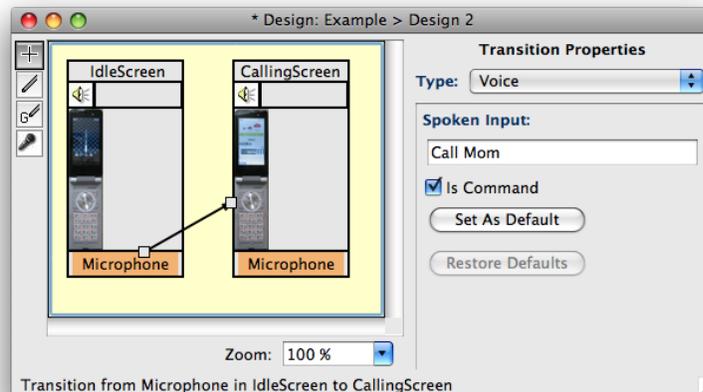


Figure 4-30: Microphone Transitions are used to enter spoken words.



CogTool does not actually “hear” the words said into the microphone. It represents a user speaking into a microphone with the contents of the Spoken Input text box on the Transition.

4.5.2 Speaker

Just as the microphone represents voice spoken by a user, the speaker represents voice or alerts uttered by the system. A speaker is an output device.



Microphone is input from the user. Speaker is output from the system.

When a speaker Device is included in a Design, every Frame has a text box at the top

that labeled with a Speaker icon. Type the text you want the speaker to output when this Frame is entered. i.e., when a Transition is followed to this Frame, in this box. As an example, when a user performs an action on the keyboard that transitions to a Frame with Speaker content, the Speaker content will be audible as soon as the transition to the Frame is made.



CogTool does not actually “speak” the words output by the speaker. It represents sounds presented to the user with the contents of the speaker text box at the top of the Frame.

The text box on the far right of the Frame is the duration of this sound. Normally, you will leave this box blank and let CogTool calculate the duration (approximately 50 ms per character). You can override this default calculation by typing a duration (in seconds) into the text box on the right of the Frame. For example, you might specify that the speaker emits “beep” (which would be estimated to be 200 ms) but you know that the beep in your system is 1 second long. You could either enter “beeeeeeeeeeeeeeeep” or enter “beep” and put 1 in the duration text box, as shown below.

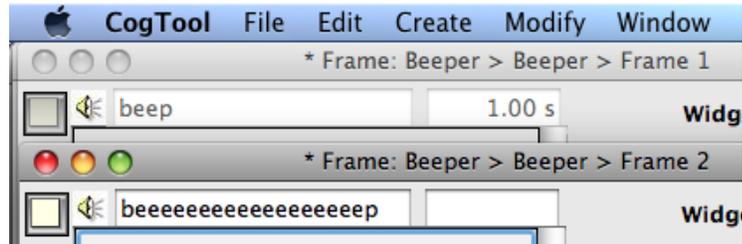


Figure 4-31: Text typed in the speaker text box is “audible” after a Transition to this Frame.

4.6 Working with HTML

4.6.1 Import a Design from HTML

If your design already exists as HTML files, either on the web or locally on your computer, you can import those files making them into CogTool Frames, Widgets and Transitions. The import function is limited, but it can be a good beginning that save a lot of effort.

You might want to do this if

- Other designers in your organization use HTML for mocking up design ideas and you want to make predictions of skilled performance time on these ideas.
- You want to compare your design to a competitors design that exists as a website.

To import a Design from HTML

- In the Project window, from the File menu, choose Import Design from HTML...
- Use the pull-down list to select a Design into which to import the pages, or create a new Design (default).
- Enter a URL to crawl, either on the web (starting with the default http://) or from your machine (replace the http:// with the path to the starting file on your machine). If you want more than one starting URL, you can add them by clicking the “+” button at the right of the field.
- Limit the import by setting the Maximum Depth to Crawl or restricting the domain, under the URL field. The small “?” buttons bring up detailed information about how to use these restriction mechanisms.



After entering the URL, you have to click on the one of the other fields in the dialog box to get the restriction paths to show their options.

- Uncheck the Capture an image for each page checkbox, if you want to decrease the size of the resulting CogTool file or speed up the import. It is checked by default because the images give you a better idea of what has and what has not been imported (and you can always remove the background images later if you need to have a smaller file).
- If desired, set the maximum number of pages to import lower than the default 500. This may be important if you have not restricted the import in any other way (i.e., by maximum depth to crawl or domain).
- If desired, set the width and height of the browser window you expect your users to see (e.g., if you are prototyping a system that would be viewed on a cell-phone browser, it would be much smaller than CogTool’s default size).
- Hit OK.

If you chose to create a new Design, the standard New Design dialog box will appear asking which Devices the New Design should include before beginning the import. If you chose to import into an existing Design, the import begins immediately and shows you a progress bar listing which page it is working on.



The number of Frames in each row of the Design window created by importing can be set in the Preferences dialog box (under the CogTool menu on a Mac; under File in Windows).

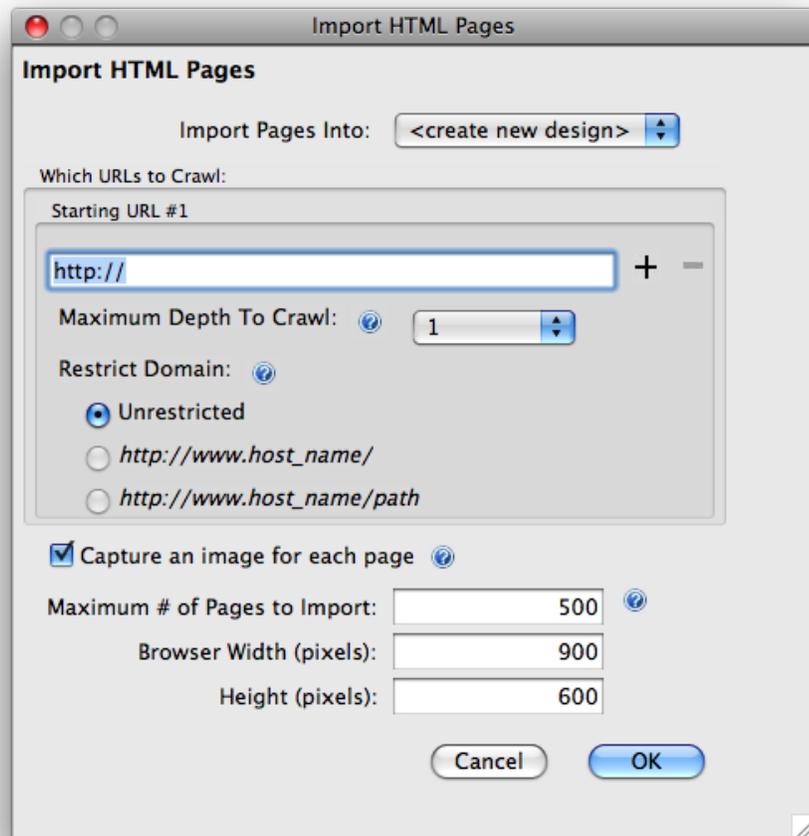


Figure 4-32: Dialog box for importing a Design from HTML. The small “?” buttons provide more detailed information about the options this dialog box provides.

The resulting Design will include the Frames and Transitions representing the pages and links in the URL(s) you imported. For example, the figure below shows the result of importing from the CogTool web site, with a Maximum depth of 1, restricted to the cogtool host.



Be careful to use the actual URL when restricting the domain. For example, the CogTool URL can be expressed as “cogtool.org” but that redirects to “http://cogtool.hcii.cs.cmu.edu/”. If you use “cogtool.org” and restrict the domain, you will only get one page because none of the linked pages are actually on that host.

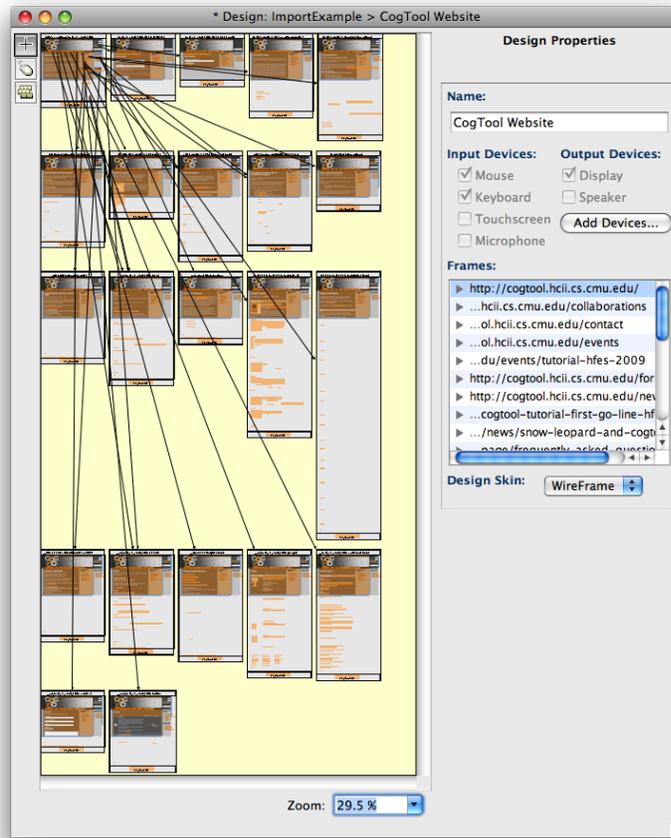


Figure 4-33: Result of importing the CogTool web site (maximum depth=1; domain restricted to <http://cogtool.hcii.cs.cmu.edu/>).

The figure below shows the Frame representing the CogTool homepage. Notice that it has Widgets (in green) on top of all the links, with the link text as the Widgets' display labels. The captured background image does not fill the Frame because the page was bigger than the browser window size (in this case, the default setting). Notice also that Widgets other than links are not represented through the import process. I would have to place Textbox Widgets over the login and search textboxes (upper right of the page) by hand to complete this Frame.

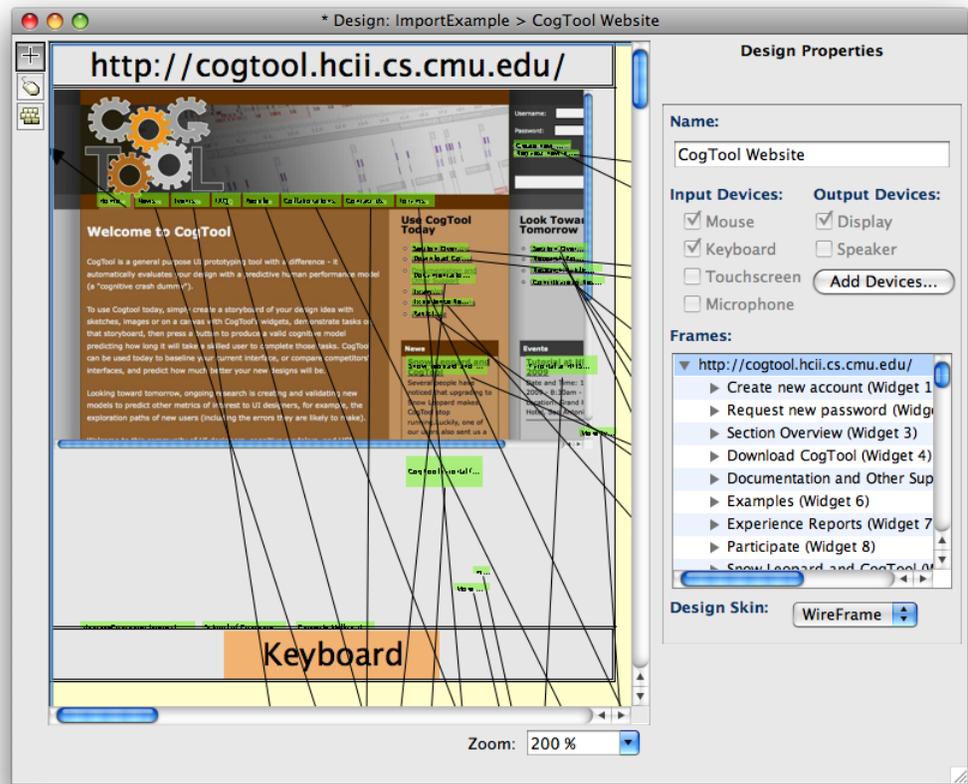


Figure 4-34: Frame representing the CogTool home page, resulting from importing the CogTool web site. Notice that (1) only link Widgets and their Transitions are imported and (2) Widgets appear in places beyond the background image, which means the user would have to scroll to see those widgets if their browser were the same size as was set in the import dialog box (in this case, the default setting).

4.6.1.1 Known problems when importing from HTML

The ability to import from HTML is useful as a start, but far from perfect. It has the following known problems.

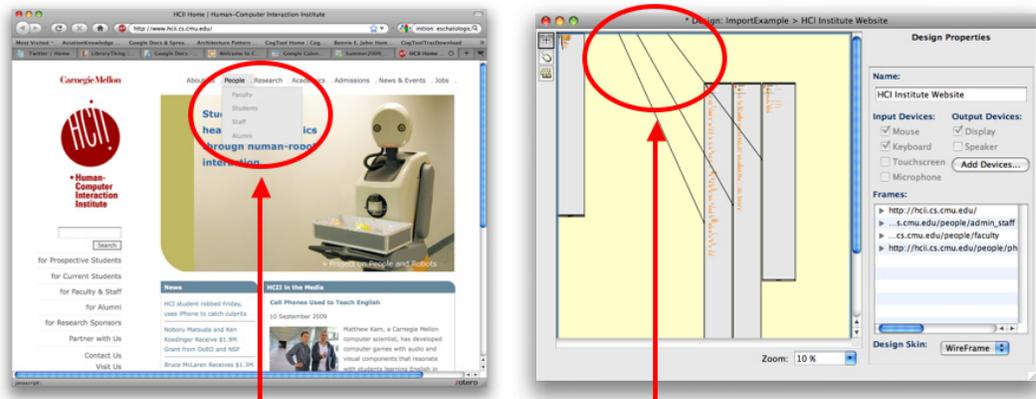
F.Y.I. If the importing process encounters any files to download or other non-HTML features, it may stop and wait for you to answer a question in a dialog box (e.g., Do you want to save or open this file?). Dismiss the dialog box by clicking whichever button tells it to ignore the question. CogTool will continue importing.

F.Y.I. Sometimes importing creates duplicate Frames of a given URL because one link goes directly there and one link was redirected from somewhere else. For example, <http://www.alumni.cmu.edu>, <http://www.cmu.edu/alumni>, and www.cmu.edu/alumni/index.html redirect to <http://www.cmu.edu/alumni/> and they all make new Frames. Just delete the duplicates in the resulting Design.

F.Y.I. CogTool cannot import from websites that require a login, even if you have already logged in.

F.Y.I.

Drop-down menus on a real webpage appear as links from nowhere when imported. You can fix this by clicking on the name of the Transition in the Design's property pain, deleting it, and creating a hierarchical menu Widget on the Frame with the drop-down menus.



Drop-down menus on the real web page appear as links from nowhere when imported

Figure 4-35: One of the problems with importing from HTML; links from nowhere result when the real web page has drop-down menus. This can be fixed, but it is one example of the imported Design being a starting place, not a perfect representation of the web site.

4.6.2 Export a Design to HTML

Export Design to HTML converts your Design into HTML pages, which can be viewed from any web browser. This allows you to share your Design with individuals who do not use CogTool or to run simple user tests where the Tasks from your Design can be explored through think-aloud usability studies.

To export your Design to HTML:

From the Project window:

- Select the Design to export.
- From the **File** menu, choose **Export Design to HTML**.

OR

From the Design window:

- From the **File** menu, choose **Export**.
- Choose **Export Design to HTML**.

CogTool saves the exported Design files in a folder of your choosing.



It is advisable to be sure the folder is empty when you export to it, or CogTool's html files will get mixed up with the files already in the folder. If you use the same folder for multiple versions of the HTML, CogTool will overwrite files of the same name add files

with new names. You can end up with several versions of screens intermingled in the same folder.

These files include HTML pages of name for each of the Frames in your Design and an index.html page from which you can navigate to these other pages (see Figure 4-36).

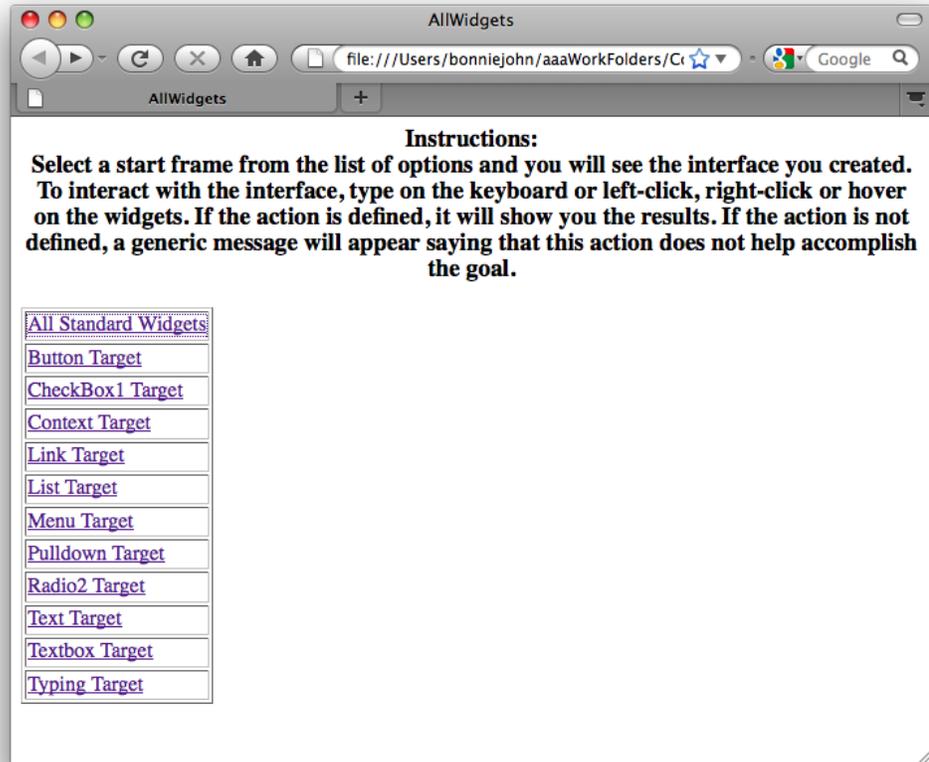


Figure 4-36: An index page of Frames exported from a Design to HTML

You can view the individual Frames from your Design by clicking the Frame name on the index page. If your Frames include interactive Widgets (e.g. drop-down menus, buttons, etc.), you can interact with the Frames directly from their corresponding HTML pages just as you would normally. For example, if you view an exported Frame with drop-down menus, simply click the menus to choose your desired option; if you view an exported Frame with a form that includes text boxes and buttons, simply type text in the boxes and click the buttons as you would normally. (see Figure 4-37).

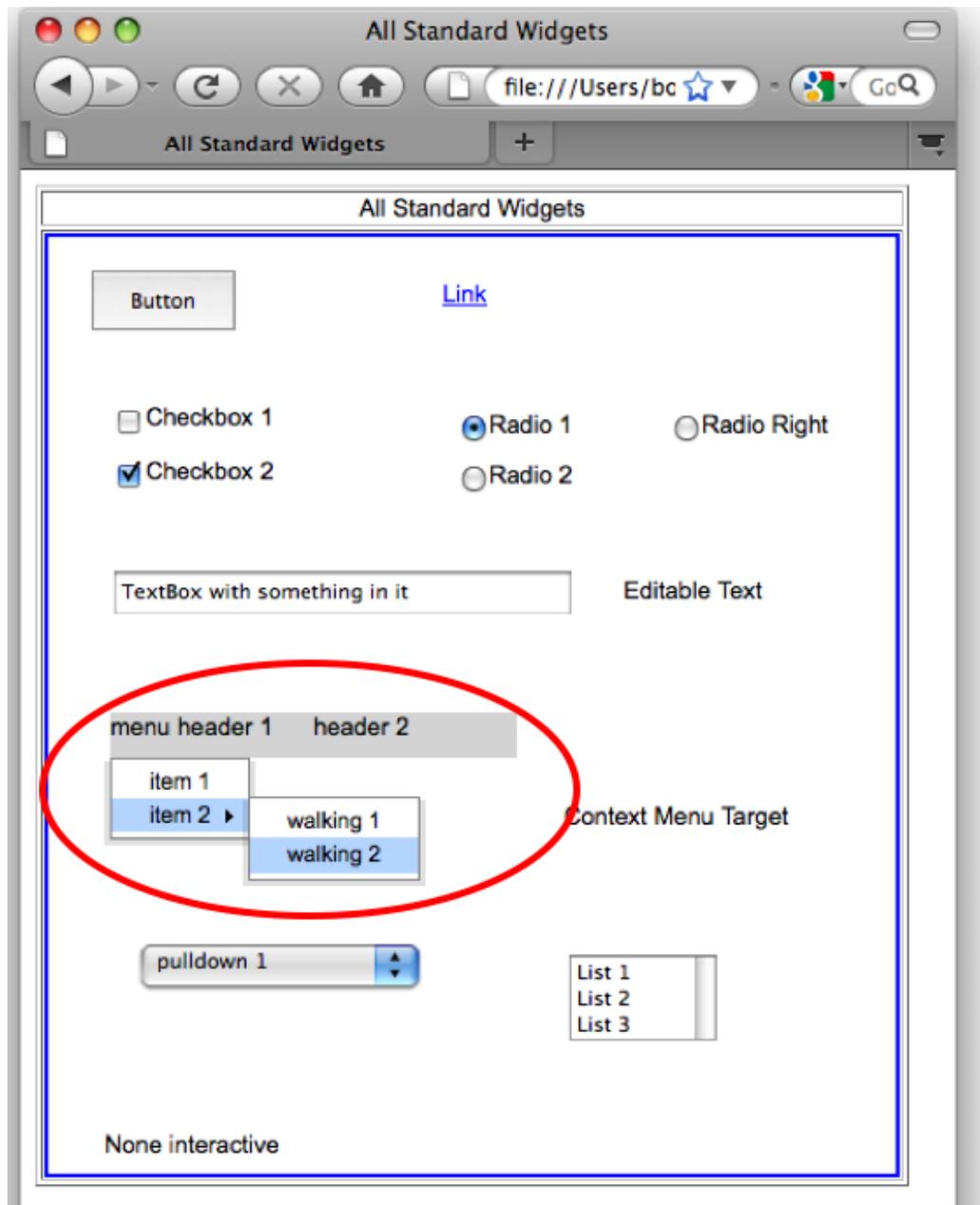


Figure 4-37: An exported Frame in HTML with all the Widgets. The drop-down menu is circled in red.

Exported HTML can be used for usability tests and two special features make them especially good for this use. First, the mouse cursor does not change when the user rolls over and active area of the screen. If it did, then people in your usability tests could use the cursor change as a clue where to click to accomplish the task, which would defeat the purpose of early usability tests. Second, if the user click an area on an HTML page that does not have a Transition defined for it, a pop-up message tells them that “This action will not help accomplish your goal”. Thus, you, as a designer, do not have to fully implement your design and the HTML will still allow you to gather data.

CogTool’s automatically-generated HTML will not suit all purposes. Previous CogTool users have used it as a starting point and edited the HTML directly to annotate it or

make it work as they want it to.



At the moment, CogTool cannot export anything sensible in HTML to simulate the use of sound in an interface.



Non-interactive widgets actually interact in the HTML. You can select and copy them, but not change them.



Standard Pulldown Lists do not export correctly. In the HTML, the header does not appear at all; only the list items appear. Thus, it is as if the first item were initially selected instead of having a header.



Standard Menu Widgets does not give the message “this will not help you accomplish your goal” if you click on a menu header that has no items defined below it.



Standard Radio button and Checkbox Widgets do not give the message “this will not help you accomplish your goal” if you click on one that doesn’t have a transition defined from it. The HTML radio buttons and checkboxes do change state, even if that state is not necessary to accomplish the task.

CHAPTER 5: QUANTITATIVE ANALYSIS

As discussed in the Overview, CogTool can make predictions of human performance given the Design and some additional information about Tasks. While CogTool's predictions of skilled execution time are to within 20% of actual user data, and are thus not intended to replace all user testing especially if more accurate time estimates are of high value to your organization, they can be used to

- Compare design ideas, allowing you to explore widely, but only conduct expensive user tests with the few most promising ideas.
- Compare competitors' products to your own during a competitive analysis or for marketing purposes.
- Provide quantitative benchmarks for internal use during development to ensure that a new design is at least as efficient as the previous version.
- Determine requirements during acquisition and test that the delivered system will meet those requirements.
- Analyze an existing system for bottlenecks in user behavior, focussing your redesign effort on the aspects of the interface that need it most.

In order to perform quantitative analysis with CogTool, you must create Tasks and Demonstrations of those Tasks on your Design. Model visualization illustrates how CogTool produced predictions about the Tasks.

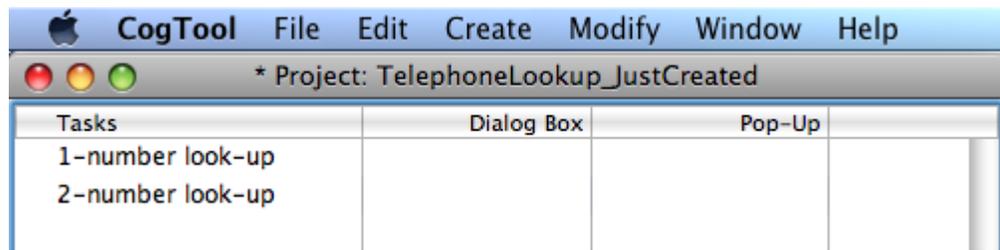
5.1 Tasks and Task Groups

5.1.1 What are Tasks and Task Groups?

CogTool uses the concept of a Task to represent the purpose of a specific sequence of actions, such as making a purchase on an online web site or setting a ring tone on your cell phone. The Task is basically a name for the sequence of steps that you Demonstrate on the Design, which CogTool records in a Script and uses to compute predictions.

A Task Group organizes the Tasks into a hierarchy, and can display the sum, mean, minimum, or maximum prediction times for the Tasks in that Group.

Tasks are created and manipulated in the Project window. The following is a Project window with two Designs and two Tasks on those Designs. These Tasks have not yet been demonstrated, so the cells at the intersections of Task and Design are still blank.



5.1.2 Creating a New Task

When you create a Project, a Task named “Task 1” is automatically inserted into the Project.

To create additional Tasks:

- From the **Create** menu, choose **New Task**.

The new Task will appear below the Task that is currently selected.

 There is no limit on the number of Tasks in a Project.

 When the new Task is created, the Task name is already selected and ready for you to type in a meaningful name. The Task name can be anything you wish using the ASCII printable character set, but it should be unique in the Task Group and descriptive of the Task. Try to use a name that will continue to apply as the Design evolves over time.

 Text in CogTool’s item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

5.1.3 Creating a New Task Group

Use Task Groups to organize your Tasks in a hierarchy. The Task Group can represent several different methods for accomplishing the same user goal (e.g., using the menu or a keyboard shortcut to invoke a command). In this case, you might want to see the mean, maximum or minimum predicted times of the Tasks that make up the Group. Task Groups can also be used to represent a higher-level goal that is accomplished by performing Tasks in the Group (i.e., subTasks). For example, to send email (the Task Group) from Gmail (the Design), you have to log in (subTask 1), compose the message (subTask 2), and send (subTask 3). In this case, you would want to see the sum of the predictions of the subTasks in the Task Group.

To create a Task Group:

- Select the Tasks that should be part of the Task Group.
- From the **Create** menu, choose **New Task Group**.

Mac: ⌘T

Win:

CTRL+SHIFT+T

Mac: ⌘G

Win:

CTRL+SHIFT+G

- The default value to be shown in the Task Group row is Sum. To change it, right-click (PC) or CTRL-click (Mac) on the word “Sum” and select the desired value in the context menu (sum, mean, minimum, maximum).

A Task Group can be comprised of Tasks and other Task Groups. There is no limit to the number of Tasks or Task Groups that may be nested within a Task Group.



When the new Task Group is created, its name is already selected and ready for you to type in a meaningful name. The Task Group name can be anything you wish using the ASCII printable character set, but it should be unique in the Project and descriptive of the group. Try to use a name that will continue to apply as the Design evolves over time.



Text in CogTool’s item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

5.1.4 Selecting Tasks or Task Groups

In the Project window, click on a the name of a Task or Task Group to select it. The entire row will turn bright blue to indicate selection.



Single-click selects a Task or Task Group and highlights the whole row in bright blue. Double-click selects the name of the Task or Task Group and highlights it for editing in light blue.

To select multiple Tasks or Task Groups

- Shift-click on each Task or Task Group you want to select. Their rows will turn bright blue to indicate selection.

5.1.5 Cutting, Copying, and Pasting Tasks or Task Groups

To copy one or more Tasks or Task Groups:

- Select one or more Tasks or Task Groups.
- From the **Edit** menu, choose **Copy Task** or **Copy Task Group**. The selected Tasks and Task Groups are on the clipboard, ready for pasting.



When you copy a Task or Task Group, it does not copy any Demonstrations or computed execution times associated with the Task or Task Group. If you want to copy the Demonstrations and predictions as well, duplicate the Task or Task Group instead.

To cut one or more Tasks or Task Groups:

- Select one or more Tasks or Task Groups.

Mac: ⌘C

Win: CTRL+C

Mac: ⌘X

Win: CTRL+X

- From the **Edit** menu, choose **Cut Task** or **Cut Task Group**. The selected Tasks and Task Groups disappear and are on the clipboard, ready for pasting.

To paste Tasks and/or Task Groups from the clipboard:

Mac: ⌘V

Win: CTRL+V

- When in the Project window, from the **Edit** menu, choose **Paste**.



A Task or Task Group can be pasted into the same Project or into a different Project.



When you paste a Task into a Task Group or Project that already has a Task of the same name, the pasted Task will be given a unique suffix (e.g. “MyTask[1]”). Likewise, when you paste a Task Group into a Project that already has a Task Group of the same name, the pasted Task Group will be given a unique suffix.

5.1.6 Duplicating Tasks or Task Groups

When you copy a Task, the Task name is copied, but the demonstrations and computed execution times are not. If you want to make an exact copy of the Task, including demonstrations and computed execution times, you should duplicate the Task.

To duplicate one or more Tasks or Task Groups:

Mac: ⌘D

Win: CTRL+D

- Select one or more Task or Task Group.
- From the **Edit** menu, choose **Duplicate Task** or **Duplicate Task Group**.



Duplicated Tasks are placed beneath the Task that has been duplicated. The Duplicated Task is given a unique suffix (“MyTask[1]”).



You can also duplicate a Task or Task Group by holding Control (PC) or Option (Mac) while dragging to a blank area of the Project window.

5.1.7 Renaming a Task or Task Group

To rename a Task or Task Group:

Mac: ⌘R

Win: CTRL+R

- Select the Task or Task Group.
- From the **Edit** menu, choose **Rename Task** or **Rename Task Group**.

OR

- Double-click on the name of the Task or Task Group.



You may name the Task or Task Group anything you wish using the ASCII printable character set. However, you can have only one Task with that name in a Task Group or at the top level of a Project. Make the name meaningful and descriptive of the Task or

Task Group.



Text in CogTool's item names must be within the ASCII printable character set. See http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters for the list of admissible characters.

5.1.8 Deleting Tasks or Task Groups

To delete one or more Tasks or Task Groups:

Mac: ⌘⌫

Win:

CTRL+DELETE

- Select the Task or Task Group.
- From the **Edit** menu, choose **Delete Task** or **Delete Task Group**.



You will be asked to confirm the deletion before the Task or Task Group is removed from the Project.



When you delete a Task Group, you will also delete all Tasks that are within that Group.

5.1.9 Changing the Order of Tasks or Task Groups

You can change the order of Tasks or Task Groups in the Project window by dragging the Tasks or Task Groups to the desired location.

You can also promote existing Tasks or Task Groups, which moves the Task or Task Group up one level in the overall schema, or demote existing Tasks or Task Groups, which moves them down one level. Promotion or demotion can move a Task or Task Group into or out of a Task Group.

To promote or demote Tasks or Task Groups:

- Select the Task or Task Group.
- To promote the Task or Task Group, press the **Tab** key (or select Promote from the Modify menu).
- To demote the Task or Task Group, press the **Shift** and **Tab** key (or select Demote from the Modify menu).

5.1.10 Changing the Display Value of a Task Group

Creating a Task Group allows you to group Tasks together and display the sum, mean, minimum and maximum times of the Tasks in the Group. You may wish to see the total time for a suite of benchmark Tasks, or you may want to see the minimum, maximum, or mean value for all the Tasks (see Figure 5-1).

To change the computed value for a Task Group:

- Select the Task Group.

- From the **Modify** menu, choose the type of value you wish to compute.

OR

- Right-click (PC) or CTRL-click (Mac) on the word label (Sum, Mean, Min or Max) and select the desired value in the context menu (sum, mean, minimum, maximum)..

Tasks	New York Guide	All Devices	New York Guide [2]
Lookup the MET	Mean: 15.239 s	Mean: 2.540 s	Mean: 15.239 s
Lookup the MET - Graffiti	15.167 s	2.756 s	15.167 s
Lookup the MET - Soft Keyboard	15.311 s	2.324 s	15.311 s

Figure 5-1: The computed value has been set to the mean and is displayed for all Designs in the Project.

5.2 Demonstrating How to Do a Task

CogTool can make predictions of task execution time for a skilled user on a given Design. To do this, it needs a Design with Frames and Transitions capable of doing the Task and a Demonstration of how to do that Task. From this Demonstration, CogTool applies an algorithm transforming the Demonstration into a Script, which is then turned into cognitively valid code in a cognitive architecture called ACT-R. This code is run in ACT-R and it produces a prediction of performance. Thus, demonstrating how to do the Task is the final step in obtaining predictions of human performance.

5.2.1 What are Demonstrations and Scripts?

A Demonstration is exactly what it sounds like, you demonstrate how to do each task on a Design. Your Demonstration is recorded in a Script, which adds some steps to your Demonstration so that it is cognitively plausible for a skilled user of your Design. For example, you may demonstrate typing “ok” on the keyboard. If the user’s hand started on the mouse, then a cognitively plausible Script contains a movement from the mouse to the keyboard (called “Home”) so the “o” can be typed with the right hand. When you demonstrate how to do your Tasks, you do not have to remember to put in steps like Home, CogTool does it for you. CogTool also adds “Think” operators in places where prior research has shown that people tend to pause before performing keystrokes or mouse actions; again, it is not necessary for you to learn the rules for placing these Think operators because those rules have been encoded into the operation of CogTool.



You demonstrate how to do a Task and CogTool creates a cognitively-plausible Script.



A Demonstration is not a user test. CogTool is not recording the time of your actions to make its predictions. You can start a Demonstration, go out for coffee in the middle, come back and

finish it, and CogTool will make the same predictions of skilled execution time no matter how long it took you to complete the Demonstration. CogTool is using prior research on human performance to make its predictions from your demonstration, not a recording of the time it took you to do the demonstration.

5.2.2 The Script Windows

There are two types of windows invoked when doing a Demonstration, one that lets you select the start Frame of the Demonstration and one where you demonstrate the task and CogTool shows you the Script it generates.

5.2.2.1 The Select the Start Frame Window

The Select the Start Frame window has two panes. The left pane lists the available Frames for that Design in alphabetical order. The right pane contains a picture of the Design with thumbnails of the Frames laid out spatially (see Figure 5-2).

If the Design's input Devices include the mouse and at least on other Device, the bottom of the left pane contains an option to select the Mouse Hand and Initial Hand Location.



When the Design has a mouse and a touch screen as input devices, CogTool does not know how to home between those two devices, nor does it know how to set the initial location of the hand correctly. Demonstrations will work, but predictions will not include the time to home between the mouse and the touchscreen Device.

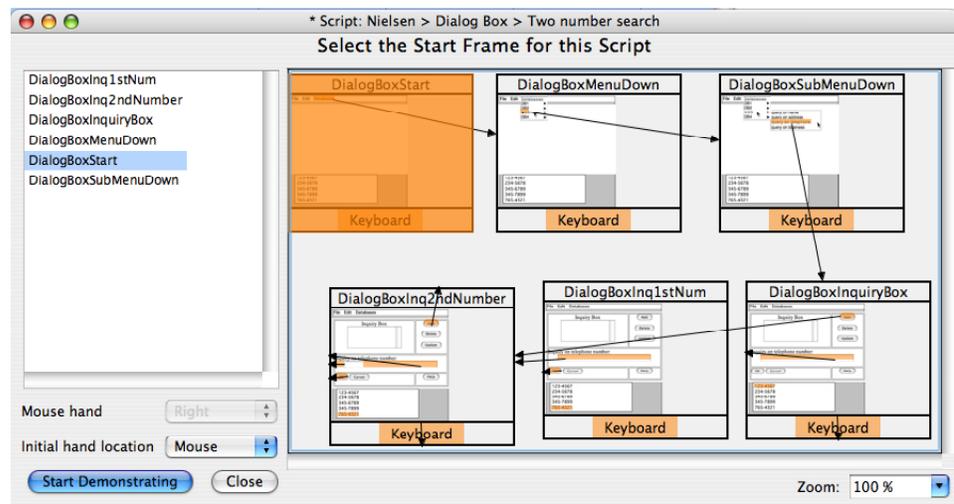


Figure 5-2: The Select the Start Frame window

5.2.2.2 The Script Edit Window

The Script Edit window appears once a starting Frame has been selected, or when a previously-demonstrated Script is opened. This window has two panes (see Figure 5-3).

The left pane shows the current Frame with its Widgets and Devices that are available for interaction.

Under the Frame are two buttons used to insert special steps into the Script. One inserts a **Look at** Widget step and the other inserts a **Think** step (discussed below).

The right pane shows the Script Step List. The steps that you have demonstrated are shown in white; steps that are logically linked to demonstrated steps are shown in gray (e.g., clicking on a menu item is logically preceded by clicking on a menu header); steps that CogTool placed automatically are shown in yellow. See the Appendix “Steps Placed Automatically by CogTool” if you are curious about how CogTool places these steps and why.

Buttons at the bottom of the right pane allow you to Delete Step and Compute. If the Design’s input devices include the keyboard and mouse or touchscreen, you can select the Mouse Hand and Initial Hand Location, in this region.

If you click on a step in the right pane, the left pane changes to show the Frame in which that step acted. You can walk through the whole task easily by clicking on the first step and using the down-arrow key to display the Frames as they would appear to the user.

Details on working with Scripts start with **Section 5.2.5 Demonstrating Actions for a Script**.

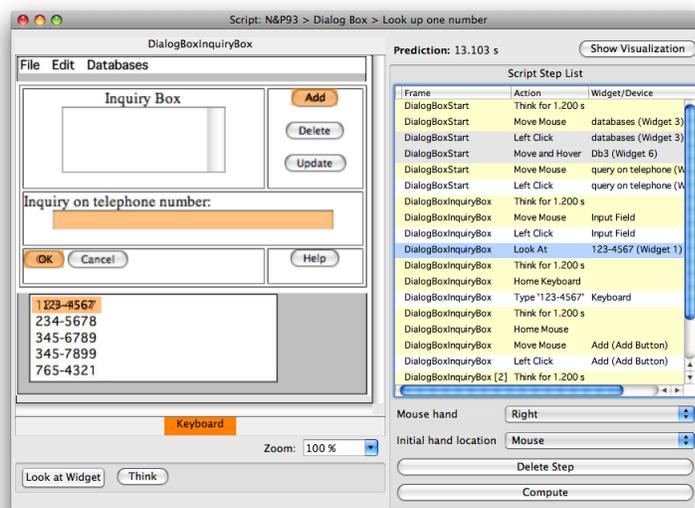


Figure 5-3: The Script Edit window

5.2.3 Opening a Script for Editing

To open or edit a Script, do one of the following:

- Click on a cell at the intersection of a Task and a Design.
- From the **Edit** menu, choose **Edit Script**.

Mac: ⌘E

Win: CTRL+E

OR

- Double-click a cell at the intersection of a Task and a Design.

5.2.4 Selecting a Start Frame for the Script

Since a Task may begin at any Frame in the Design, you must select the Start Frame for each Script. The first time you open a Script, you will be presented with the Select the Start Frame window.

To select the starting Frame, do one of the following:

- Double-click on the Frame in the right pane or the Frame name in the left pane

OR

- Choose the Frame from either pane, then click the Start Demonstrating button.

If the keyboard and the mouse or touchscreen are input devices in your Design, buttons at the bottom of the left pane will allow you to choose the Mouse Hand and the Initial hand location. These options are also available in the Demonstration Script window.



When the Design has a mouse and a touch screen as input devices, CogTool does not know how to home between those two devices, nor does it know how to set the initial location of the hand correctly. Demonstrations will work, but predictions will not include the time to home between the mouse and the touchscreen Device.

Once the starting Frame has been selected, the Script Edit window will appear. The starting Frame appears in the left pane of the Script Edit window and the current list of steps appears in the right pane. At this point, the list of steps only includes the name of the starting Frame.

Zoom In

Mac: ⌘=

Win: CTRL+=

Zoom Out

Mac: ⌘-

Win: CTRL+-

Normal Zoom

Mac: ⌘0

Win: CTRL+0

Zoom to Fit

Mac: ⌘/

Win: CTRL+/

You can use the same zoom commands that you used when you created and edited the Frames to allow you to see the Frame better.

To use the zoom commands, do one of the following:

- From the Modify menu, choose the Zoom option you want.

OR

- Choose your zoom settings from the box at the bottom of the Frame window.



Normal Zoom makes the contents of the frame revert to the default size. **Zoom to Fit** makes the contents of a frame fit the available space in the window.

5.2.5 Demonstrating Actions for a Script

5.2.5.1 Simple Demonstration

You can record steps for the Script by interacting with the Frame in the left pane in a way similar to how a user would interact with the actual device. Assuming you have added all the Devices, Widgets and Transitions needed to demonstrate the Task, you just left-click on the appropriate Device or Widget to make the action associated with left-click occur. If a left-click Transition is defined on that Device or Widget, CogTool follows that Transition to the next Frame. If no left-click Transition is defined on that Device or Widget, a context menu pops up from which you can select a different previously defined Transition, define a new self-Transition, or Look-at the Widget (discussed below).

If there is a left-click Transition defined on the Widget, but you do not want to follow that Transition at this point in the task (e.g., there is also a right-click Transition defined and that is the one you want to follow), right-click (PC) or CTRL-click (Mac) on the Widget and the context menu will appear, giving you access to all the pre-defined Transitions as well as the option to define a new self-Transition or Look-At the Widget (discussed below).



If you have difficulty remembering which Widgets have Transitions or actions to trigger them, hover the mouse over the Widget to display a tool tip. The context menu for the Widget will also have information about Transitions



When a left-press Transition is defined on the Widget, CogTool follows the left-press Transition when you left-click on the Widget. CogTool is catching the “press” before it sees the “click”, while understandable, but not as useful as having it see a click as a click. If a Widget has both left-press and left-click Transitions defined on it, access the left-click through the context menu.

In the case of a touchscreen interface, CogTool automatically interprets left-clicks as taps, e.g., like a tourist guide on a iPhone with a Museums button (see Figure 5-4).

Frame	Action	Widget
1-Start	Think for 1.2 s	
1-Start	Move and Tap	Museums
2-Museums_List		

Figure 5-4: CogTool demonstrating a touchscreen action. You, the analyst, clicks on the Museum button, but CogTool knows you have prototyped a touchscreen interface and records a tap, not a click.

When you click on Widgets that have parts that are not visible when the Frame is entered, e.g., menu, context menu, and pull-down list Widgets, they open and you can continue clicking until you get to the Widget that emits the next Transition. CogTool puts all the steps that are necessary to mimic the complex interaction into the Script (in gray), as defined by the Widget when it was created. For example, if you defined a menu system that used hover to open submenus with a delay of 0.5 sec (typical for a PC), you will demonstrate by clicking on the menus and submenus, but the CogTool Script will list the hovers and system waits as defined.



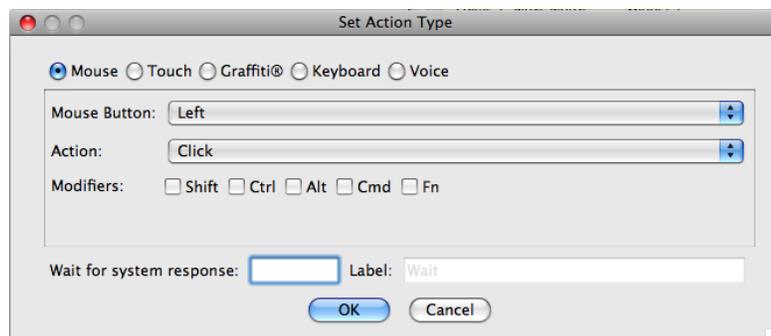
If your Frame has overlapping Widgets, a left-click will automatically follow the Transition for the topmost Widget. Actions on all occluded Widgets are accessible through the context menu.

5.2.5.2 Here's Why Using the Right Widgets was Important

CogTool creates cognitively-plausible Scripts by adding steps to your Demonstration. Some of these steps are physical, like homing the hands between the mouse and the keyboard. But most of them are Think steps, placed where they are because prior research has studied where people pause when using computers (e.g., Card, Moran, and Newell, 1980; Lane, et. al., 1993). CogTool employs a mapping of the original rules for placing mental operators that allows it to connect Think steps to Widgets under certain circumstances. The rules for placing Think steps are too complex to document here, but you can see that using some Widgets under some circumstances places Think steps and using other Widgets do not. The number of Think steps will impact the prediction of skilled execution time, so using the right Widgets when creating your Design is important. (See the Appendix “Steps Placed Automatically by CogTool” if you are curious about how CogTool places Think steps and why.)

5.2.5.3 Adding Self-Transitions During Demonstration

If you discover that you forgot to create a self Transition in your Design, you can add a self-Transition step on any Widget in any Frame during Demonstration. Simply use the context menu on the desired Widget and select Perform Self Transition. A dialog box will appear that lets you add any available Transition for that Widget. This will create a new step in the Script, just as if you had put the Transition in the Design prior to demonstrating the task. It will not, however, actually put the self-Transition into the Design, so if you think you will need it again, you should go back to the Design and add it.



5.2.5.4 Adding Look-At Steps During Demonstration

When computing, CogTool will automatically produce a model of human performance that looks at the Widgets it clicks on as it does the task. All you have to do to produce such a model is demonstrate how to do the task as described in previous sections. However, there are also times when users will have to look at Widgets more deliberately, e.g, to get information from a Non-Interactive Widget to make a decision about which Button Widget to press, or to read a dialog box that comes up with new information each time. You can simulate this in CogTool by adding a Look-At Widget step as you demonstrate the task.

To add a Look-At Widget step during demonstration

- Click the Look-At Widget button at the bottom of the Script window. It will toggle and look selected.
- Click on the Widget you want the user to look at, comprehend, and deliberate about. The Look-At Widget button will toggle back to not-selected and a deliberate Look-At step will be inserted into your Script.

OR

- Use the context menu on the Widget and select Look-At Widget.

CogTool assumes that if you have inserted a Look-At step deliberately, a skilled user would be doing this to gather and comprehend information, and therefore be thinking after doing the Look-At step. Therefore, CogTool places a Think step after every Look-At step you demonstrate.



Deliberately looking at widgets should be used sparingly. You do NOT need to use them just to look at a button to click on it; CogTool does that for you when it computes predictions. However, if your user will need to look around at different parts of the display to gather information with which to decide what to do next (e.g, a commercial airline pilot looking at specific areas of the cockpit to know which buttons to press to implement course changes from an Air Traffic Controller), use Look-At Widget in your Demonstration.

5.2.5.5 Adding or Removing Think Steps During Demonstration

In some cases, you may wish to change the “Think” steps that CogTool automatically inserts in the Script. Because the rules CogTool uses to place “Think” operators have been validated through many experiments spanning several decades, changing the placement of Think steps is discouraged unless you have empirical evidence applicable to your Design to support the change.

To remove a “Think” operator:

- Select the line in the Script with the “Think.”
- Click the Delete Think button at the bottom of the pane.

Conversely, if you believe that you have sufficient justification, you may add additional Think steps.

To insert a Think step during Demonstration:

- Click the Think button at the bottom of the Script window. A dialog box will appear asking you the duration of the Think (based on your data or other literature that pertains to your Design) and whether you want to label the Think in some way specific to your Design. For example, you may know that it takes 10 seconds to read a dialog box that appears in your design. That would be a justifiable reason to add a Think step to the Script, as shown.



- The Think step will be added at that point in the Script.



Numbers can only be entered into CogTool in US format, i.e., using digits and a “.” to represent the decimal point.

To insert a Think step after Demonstration:

- Select the step in the Script that you want to put a Think before.
- Click the **Think** button.
- In the dialog box that appears (the same as shown above), enter the amount of time in seconds the user will pause to think.
- The Think step will be inserted above the step you have selected. If the step you selected was a demonstrated step that had automatically-generated steps before it (e.g., homing), the Think step will appear before all the generated steps.



Because of prior research, CogTool will not allow a Script to have two Think steps in a row. If you add a Think step in a position where it will be next to an automatically-generated Think step, CogTool will replace the automatically-generated Think step with your deliberately-added Think step.



You can also change the duration and label of any Think step at any time by double-clicking on it in the Script window. The dialog box for changing the duration and label will appear.

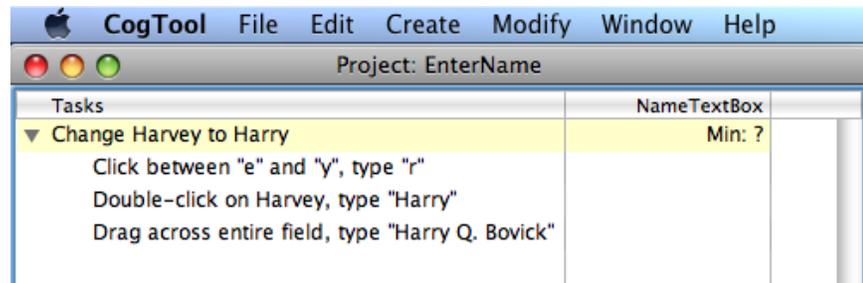


Numbers can only be entered into CogTool in US format, i.e., using digits and a “.” to represent the decimal point.

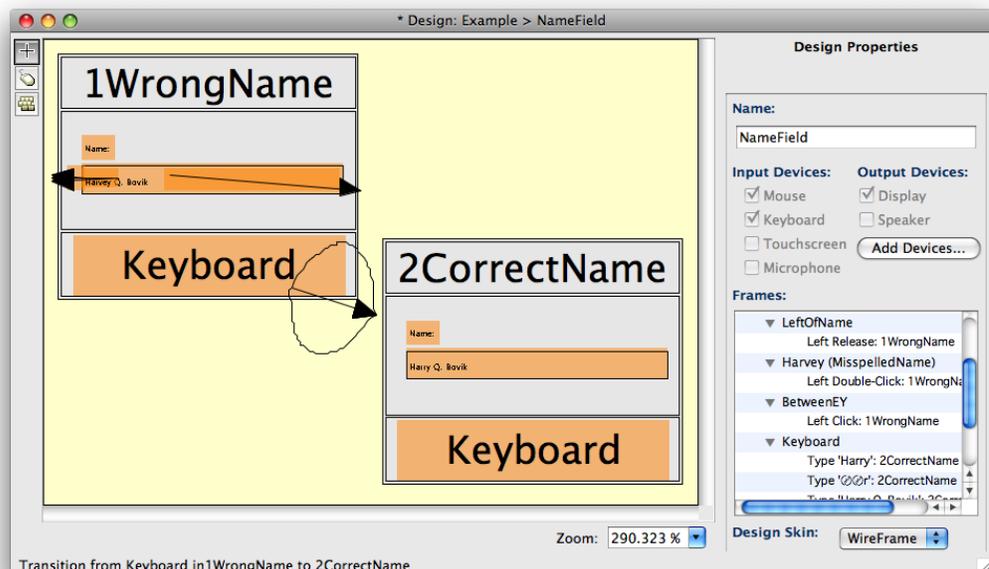
5.2.5.6 Examples: Demonstrating the Three Text-Editing Tasks Designed in Previous Sections

In several previous sections (see “Working with Text Boxes & Text” and “Dragging Over Text to Highlight it” in the previous Chapter), we have discussed editing text to change a name, “Harvey Q. Bovik” to “Harry Q. Bovik”. We built a Design with the Frames, Widgets and Transitions to simulate three methods for accomplishing this task (1) positioning the cursor between the “e” and “y” in Harvey, backspacing twice and then typing “r”, (2) double-clicking on “Harvey” to highlight it and then typing “Harry”, and (3) dragging across the entire field to highlight it and then typing “Harry Q. Bovik”. The following subsections show pictures of how each of those method would be demonstrated.

The Project is set up with a Task Group containing the three methods of editing the text.



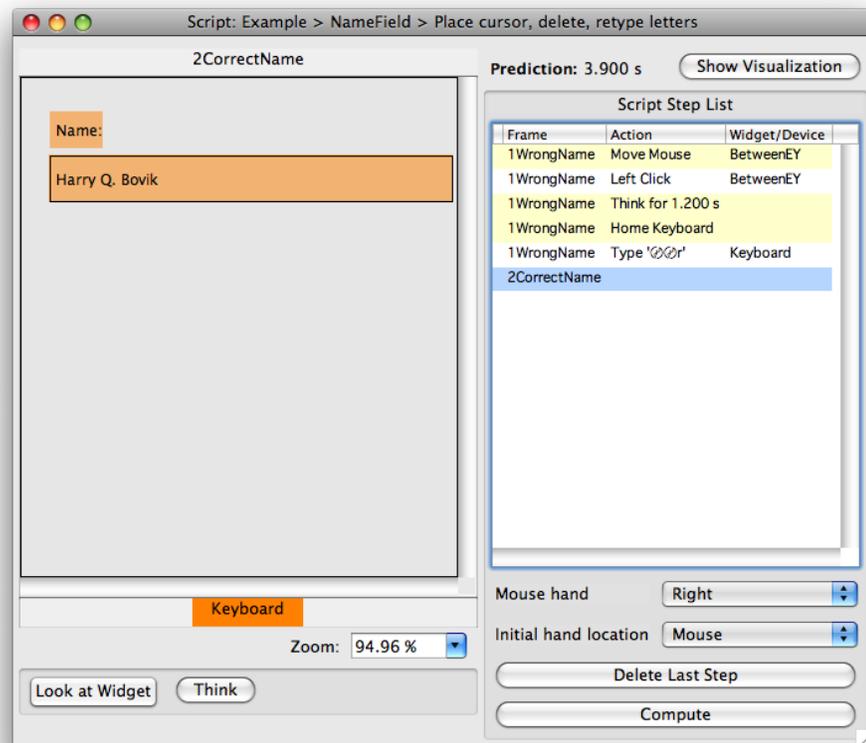
The Design contains all the necessary Frames, Widgets and Transitions to demonstrate all three methods. Notice that the Transitions that highlight the text are all self-Transitions and that there are multiple keyboard Transitions to the next Frame because the different methods require the user to type different things on the Keyboard. You can access any of these transitions, even if they overlap with others, by clicking on their names in the Design's properties pane.



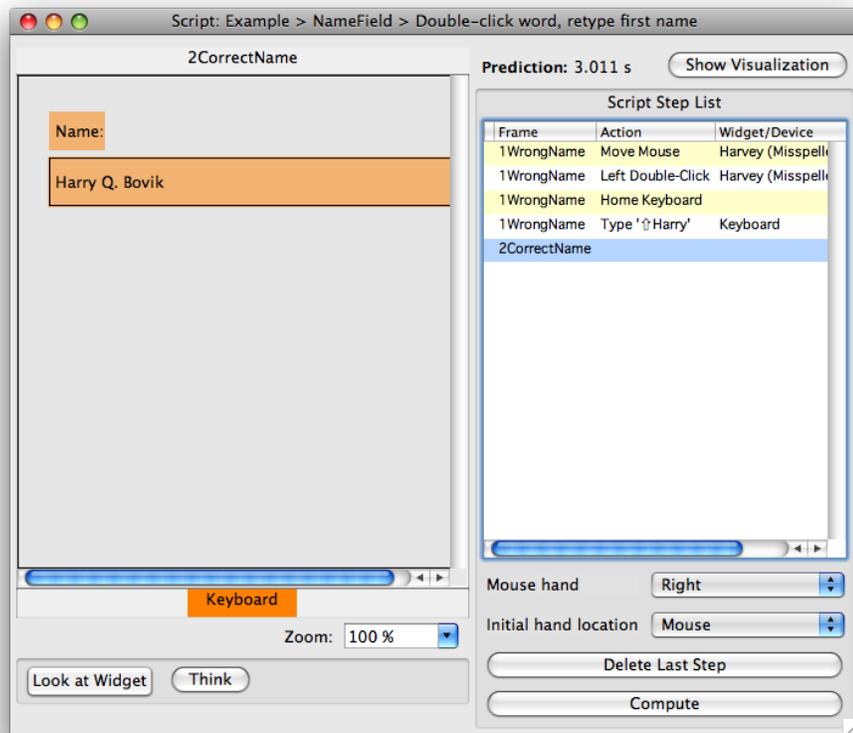
The Transitions are:

- For the first method: left-click self-Transition on text Widget between the “e” and the “y”. The keyboard Transition to the next Frame contains “backspace backspace r”.
- For the second method: left-double-click self-Transition on Harvey. The keyboard Transition to the next Frame contains “Harry”.
- For the third method: left-press self-Transition on Right_of_Name and left-release self-Transition on Left_of_Name. The keyboard Transition to the next Frame contains “Harry Q. Bovik”.

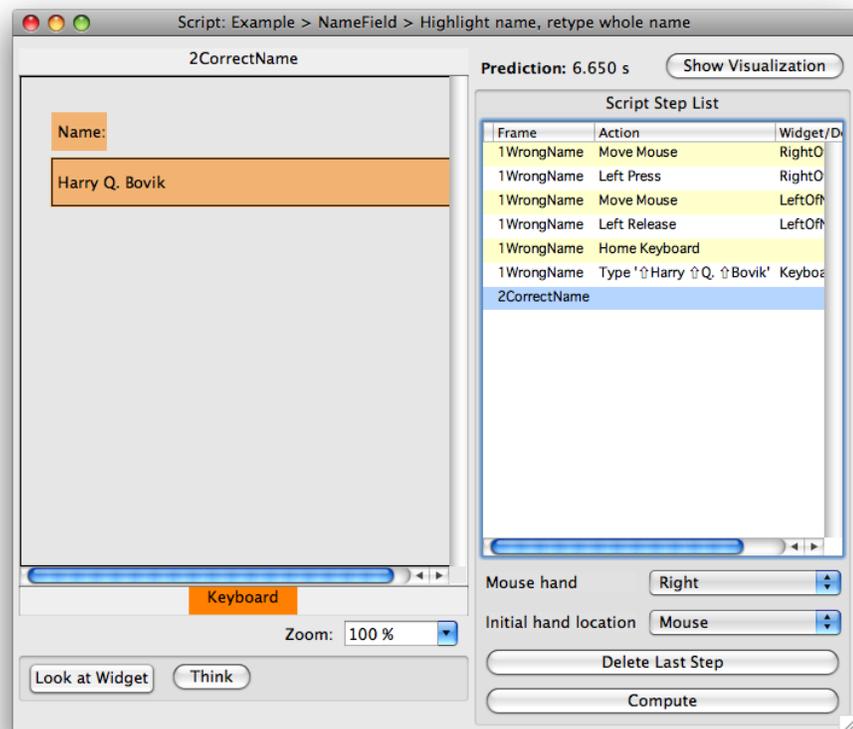
The Script for the first method, clicking between e” and “y” in Harvey, backspacing twice and then typing “r” is:



The Script for the second method, double-clicking on “Harvey” to highlight it and then typing “Harry” is:



The Script for the third method, dragging across the entire field to highlight it and then typing “Harry Q. Bovik” is:



Below, you can see the Project window that compares these three predictions (after they have been computed, see the section on computing predictions). CogTool predicts that double-clicking and retyping the first name will be the fastest. This makes sense because placing the cursor between two letters is a much more precise movement than double-clicking on a word and the delete key is further away from the home-row than the letters of “Harry”. However, remember the warning about CogTool’s typing predictions that they are unreliable when the task is very short. Therefore, these predictions should be interpreted to mean that the third method, highlighting and retyping the entire name, is certainly slower than the other two methods, but the first two methods are not reliably different.

Tasks	NameField
▼ Fix Name	Min: 3.011 s
Place cursor, delete, retype letters	3.900 s
Double-click word, retype first name	3.011 s
Highlight name, retype whole name	6.650 s

5.2.6 Editing Think Script Steps

You may edit **Think** steps in your Demonstration by double-clicking them and then editing the duration and label. You may not directly edit any other steps in a Script.

5.2.7 Deleting Script Steps

It is not advisable to delete arbitrary steps in the middle of a Script. Although CogTool does not prevent you from deleting an arbitrary step, the resulting Script is not guaranteed to produce a psychologically valid prediction. It is better to delete steps from the end of the Script, as follows.

To delete script steps from the end of the Script:

- Select the last step you demonstrated.
- Click **Delete Step** until you have deleted back to the undesired step. As you delete each demonstrated step (in white) all generated steps associated with it (in yellow or gray) will be deleted as well.
- Restart your Demonstration from the deleted step.



You will not be able to delete steps that are generated by a Speaker device in this window. If you double-click on a Speaker step, a dialog box will instruct you to edit the Frame containing the Speaker information and give you a button to take you to that Frame. You can get to the Frame at any time by double-clicking on its title in the left pane of the Script window.

5.2.8 Reviewing Your Scripts

5.2.8.1 Reviewing the Steps in a Task Script

When you click on the Script step in the Script list, the associated Frame appears in the left pane of the Script Edit window. You can review how to do the Task by clicking on the first step in the list and using the arrow key to walk through the steps, showing all the Frames as you arrow-down.

If you need to change something, you can easily get to any Frame window by double-clicking its name in the left pane of the Script window.

5.2.8.2 Reviewing the Steps in a Summed Task Group

If you have a Task Group that represents a higher-level goal that can only be completed by completing all the subTasks (i.e., the Task Group is set to display the sum of the predicted times of the Tasks). You may want to see all the steps in all the Tasks in one window, and CogTool provides this ability.

To view all the Steps for such a Task Group:

- Double-click the cell which contains the summed value for the group.

You cannot edit the Scripts in this window, only review them. Open the individual Task's Scripts to edit them.

5.3 Computing a Prediction

Once you have finished creating and editing a Script, click the Compute button at the bottom of the right pane to compute a prediction. The result will appear as a value at the top of the right pane in the Script window (to the right of the “Prediction” label) and also in the Script cell for the Task in the Project window.

The number that appears in these locations is the calculated prediction for the execution time of a skilled user. If you do not have numbers in cell in the Project window, you might have one of the following:

- Nothing (): the cell has not been demonstrated and lacks a Script.
- Double dash (--): the Script has been demonstrated but not yet computed.
- Question marks (??): there is an invalid step, such as clicking on a Menu Item Widget before clicking on a Menu Widget when using Custom Widgets. The Script steps affected will be colored red and will be marked by an X on the left-hand side of the “Script Step List.” The Script will need to be edited before it can be recomputed (see Figure 5-5). This situation almost never occurs when you use Standard Widgets in your Design, but it is a common error when you use Custom Widgets.

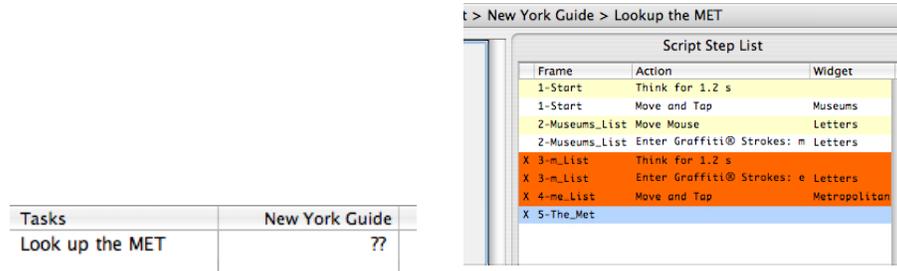


Figure 5-5: Question marks in the Script cell and red lines in the “Script Step List”

- **Double-X (XX):** a problem exists with a Transition that causes the prediction time for the Script to be greater than 600 seconds (10 minutes). We have set 600 seconds as a maximum time in case CogTool gets into an infinite loop. This rarely happens on .cgt files created in the same version of CogTool in which you are trying to compute (it does occasionally happen when running an old .cgt file in a new version of CogTool), but if it does, it usually means there is something wrong with your Script. If your Script is correct but your system includes tasks that will take longer than 600 seconds, break them up into smaller pieces in and organize them into Tasks and Task Groups (or contact us to make a special CogTool for you that can run a longer simulation).



CogTool is known to have difficulty computing when using it in non-English contexts on a Mac, particularly those that use non-Roman scripts in their input methods, as set by the International System Preferences.



Keyboard transitions that involve many special characters in a row result in predictions of human performance that are slower than the performance of actual skilled computer users. This is because (1) KLM assumes all keystrokes are done in sequence even for keys held down together and (2) ACT-R (the cognitive architecture making the predictions “under the hood”) provides a model of typing that is too simple, assuming each finger has to move back to the home-row before typing the next key. For most comparisons between interfaces, these assumptions are reasonable approximations to user behavior. However, for extremely skilled users of multiple-key keyboard shortcuts and navigation keys, the predictions will be too slow.

5.3.1 Recomputing Scripts

When you make changes to your Script, it will automatically regenerate. However, you should always recompute a changed Script to ensure an accurate, updated prediction time.

To recompute a Script:

From the Script window:

- Click the **Compute** button.

From the Project window:

- Select the Script cell.
- From the **Modify** menu, choose **Recompute Script**.



You can recompute all Scripts in a Project window by selecting all the Scripts. To select all the Scripts, from the **Edit** menu, choose **Select All**.

Mac: ⌘A

Win: CTRL+A



Some changes to a Design can prevent a Script from being recomputed properly, e.g., deleting a Frame or Widget that is used in the Script. Invalid Scripts must be re-demonstrated. The red areas in the Script window indicate from which point the associated Script must be re-demonstrated. If the Recompute Script command produces an erroneous result, you may need to re-demonstrate the Script from the beginning.

5.4 Exporting Scripts and Results to CSV Files

Some CogTool users have found exporting Scripts and results to CSV (comma separated value) files useful and we present how to do that here. You can also export designs to HTML, which is discussed at the end of the last chapter.

5.4.1 Exporting a Script to CSV

You may want to export your Script to a CSV file to view the steps in a more concise and manipulable format than the Script windows, or to use them in reports about your analyses. Usually, you will want to open the CSV file in a spreadsheet program, such as Microsoft Excel or Google Docs.

To export a Script:

- Select the Script cell from the Project window, or open the Script window.
- From the **File** menu, choose **Export Script to CSV**.

The default file name for the exported Script follows the standard of **Project Name_Design Name_Task Name.csv**.

The CSV file contains information on the version of CogTool, the Project and Design names, and all the information from the Script window (the Frames, Actions, Widget Names and Types, and Displayed Labels) (see Figure 5-6).



When you export a Script that is a member of a Task Group, the Task Hierarchy is displayed under the Design name (see Figure 5-6).

	A	B	C	D	E	F
1	Date and Time:	6/25/08 11:23				
2	Project Name:	Tutorial Project				
3	Design Name:	New York Guide Tutorial				
4	Task Hierarchy:	Lookup the MET	Lookup the MET - Graffiti			
5						
6	Frame	Action	Widget-Name	Displayed-Label	Widget-Type	
7	1-Start	Look At	Museums		Button	
8	1-Start	Think for 1.200 s				
9	1-Start	Move and Tap	Museums		Button	
10	2-Museums_List	Move Mouse	Letter Input Area		Graffiti/Æ	
11	2-Museums_List	Enter Graffiti/Æ Strokes: m	Letter Input Area		Graffiti/Æ	
12	3-m_List	Enter Graffiti/Æ Strokes: e	Letter Input Area		Graffiti/Æ	
13	4-me_List	Move and Tap	Metropolitan Museum		List Box Item	
14	5-The_Met					
15						
16						
17						

Figure 5-6: The information from the Script window exported into Microsoft Excel



When you open the file in a spreadsheet application, such as Microsoft Excel, symbols for the special keys (e.g. ↑ for Shift) may not be displayed. In these cases, the symbols are replaced by the key name, which is capitalized and in parentheses (e.g. (SHIFT)) (see Figure 5-7).

	A	B	C
1	Date and Time:	6/27/08 11:06	
2	Project Name:	Untitled Project 3	
3	Design Name:	Design 1	
4	Task Hierarchy:	Task 1	
5			
6	Frame	Action	Widget-Name Disj
7	Frame 1	Think for 1.200 s	
8	Frame 1	Type '(SHIFT)'f	Keyboard
9	Frame 1	Think for 1.200 s	
10	Frame 1	Say 'go'	Microphone
11	Frame 1	Think for 1.200 s	
12	Frame 1	Type '(CTRL)'s	Keyboard
13	Frame 1		
14			

Figure 5-7: An exported Script with special keys: For a complete list of the special keys and their corresponding symbols, see **Appendix: Special Keys and Symbols**.

5.4.2 Exporting All Results to CSV

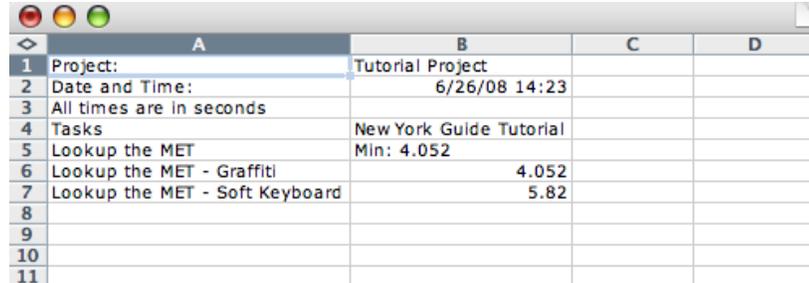
You can export all the results shown in the Project window to a CSV file. You might choose this function to manipulate (add, subtract, etc.) numbers in Microsoft Excel or Google Docs or to more easily track and work with many Tasks and Designs instead of viewing them in a large CogTool file, or to insert them into reports.

To export all results to CSV:

- Display the Project window.
- From the **File** menu, choose **Export All Results to CSV**.

The default file name for the exported script follows the standard of **Project Name_Date_Time.csv**.

The CSV file contains information on the version of CogTool, the Project Name, the date and time stamp, and all the information from the Project window (the Tasks and the computed times, in seconds) (see Figure 5-8).



The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C	D
1	Project:	Tutorial Project		
2	Date and Time:	6/26/08 14:23		
3	All times are in seconds			
4	Tasks	New York Guide Tutorial		
5	Lookup the MET	Min: 4.052		
6	Lookup the MET - Graffiti	4.052		
7	Lookup the MET - Soft Keyboard	5.82		
8				
9				
10				
11				

Figure 5-8: The information from the Project window exported into Microsoft Excel. This Project had one Design, one Task Group with the minimum time displayed and two Tasks in the Task Group.

5.5 Model Visualization

The underlying computation cognitive engine that makes predictions of human performance in CogTool is called ACT-R. It is complex and well validated with decades of research. To learn more about it, visit the ACT-R website: <http://act-r.psy.cmu.edu/>.

We have built a visualization tool into CogTool so that you can see what ACT-R does to produce the predictions. Although this visualization is meant for a cognitive psychology researcher to validate CogTool's and ACT-R's predictions, it is fun and you may find it interesting. Some designers have even found that it helps them identify bottlenecks in task execution, understand why one Design is faster than another, or inspire new Designs.

To see the visualization:

From the Project window:

- Select the Script cell of interest.
- Right-click (Windows) or CTRL-click (MacOS) the cell to bring up the context menu.
- Choose **Show Model Visualization**.

OR

From any Script window:

- Click the Show Visualization button (see Figure 5-9).

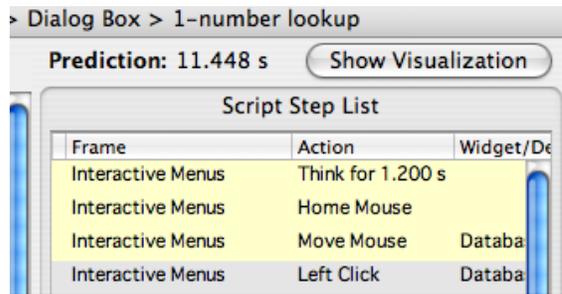


Figure 5-9: The Show Visualization button in the Script window

The Visualization window has two panes (see Figure 5-10). In the left pane is a timeline showing the different activities ACT-R goes through to make the predictions. The left pane is split into two, with the top portion of the pane presenting the entire model in miniature and the bottom portion showing a blow-up of a portion of the timeline selected from the top portion. The right pane shows either single operator information or a textual trace of the ACT-R model's run, depending on whether the **Trace** or **Operator Info** tab is selected.

In the left pane, time runs from left to right and the widths of the boxes are proportional to the time they take to execute. The rows of boxes are different types of “operators” that happen in the course of performing these Tasks.

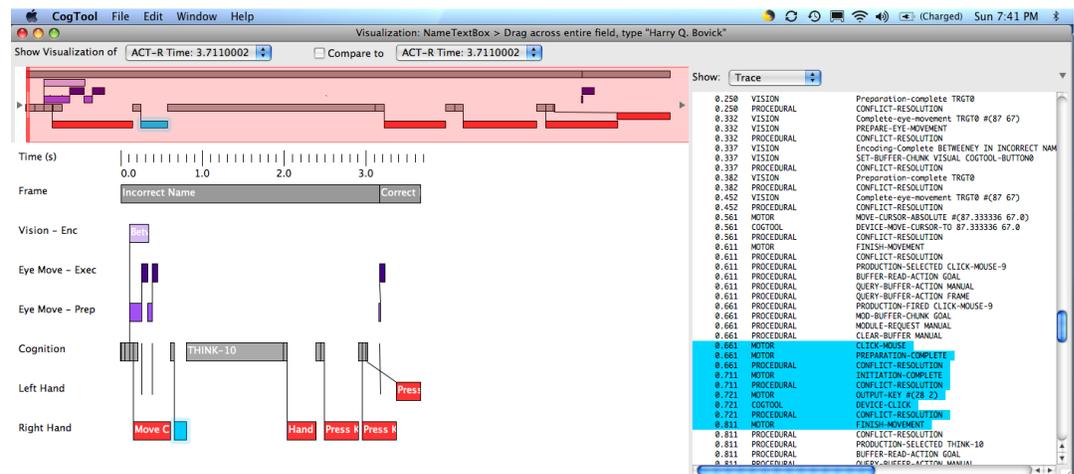


Figure 5-10: The Model Visualization window

The operators are grouped into four categories:

- System - (shades of gray)
 - * Frame (darker gray) - represents the Frame that is visible on the screen.
 - * Wait (lighter gray) - represents the system response time.



The Frames are not actually visible as long as these boxes indicate. When there is a System Wait, one Frame disappears and the next one does not show up for the duration of the System Wait. This is not shown correctly in the current visualization.

- Eyes - (shades of purple)
 - * Vision - Enc (light purple) - represents the eyes encoding objects on the Frame, i.e., putting them into a representation that ACT-R's cognitive processes can work on them
 - * Eye Move – Exec (darkest purple) - represents the eyes moving to objects on the screen
 - * Eye Move – Prep (mid purple) - represents the eyes preparing to move to objects on the screen.
- Cognition - center row (gray) - represents the thoughts the model has when performing the Task. The long boxes in this row are the Think steps. Short boxes are other types of cognitive operators that initiate motor movements and visual attention shifts.
- Hands - (red) represent motions of the left hand and right hand, as labeled.
 - * Shows the motor movement that would be observable on a video: horizontal movements of a finger, stylus, or mouse; clicks of a mouse button; or key presses. In this example, The right hand moves the mouse and clicks in between the “e” and “y” in Harvey, then the right hand moves from the mouse to the keyboard so it can type Backspace, Backspace and then the left had types “r”. (see Figure 5-10).



Hand movements on a touch screen should be labeled “Move-finger movement” but they currently appear in both the chart and the Trace as “Move-Mouse”. The calculations are correct, its just the labels that are not.



If you are trying to visualize a old .cgt file from a previous version of CogTool, the rows sometime comes out in a different order that presented here and many lines cross each other. If this happens, try recomputing and re-visualizing.

If you click on a box in the detail pane, it highlights the box in bright blue and either gives information about the operator you just selected (in the Operator tab) or highlights the part of the model trace that corresponds to it in the right pane (in the Model Trace tab).

The area shaded in light red on the overview time line represents what is currently showing in the detailed time line. The detailed time line can be manipulated by moving the selection box on the overview time line. You can re-size and move the red bars at the ends of the red area. You can move the whole area by clicking the red area and dragging it, or you can move one end by clicking and dragging a vertical end bar. You can also scroll smoothly by pressing down on the arrows (◀ or ▶) at the edges of the overview timeline, or you can step through by clicking on those arrows.

The figure below (see Figure 5-11) shows a comparison between two different models for changing “Harvey” to “Harry”. The top shows clicking between the “e” and the y” and then typing backspace, backspce, “r”. The other shows dragging over he entire text box and typing “Harry Q. Bovik”. The top model takes less time, but requires more

thinking time to remember the function of the backspace key. The bottom model takes more time because of the typing.

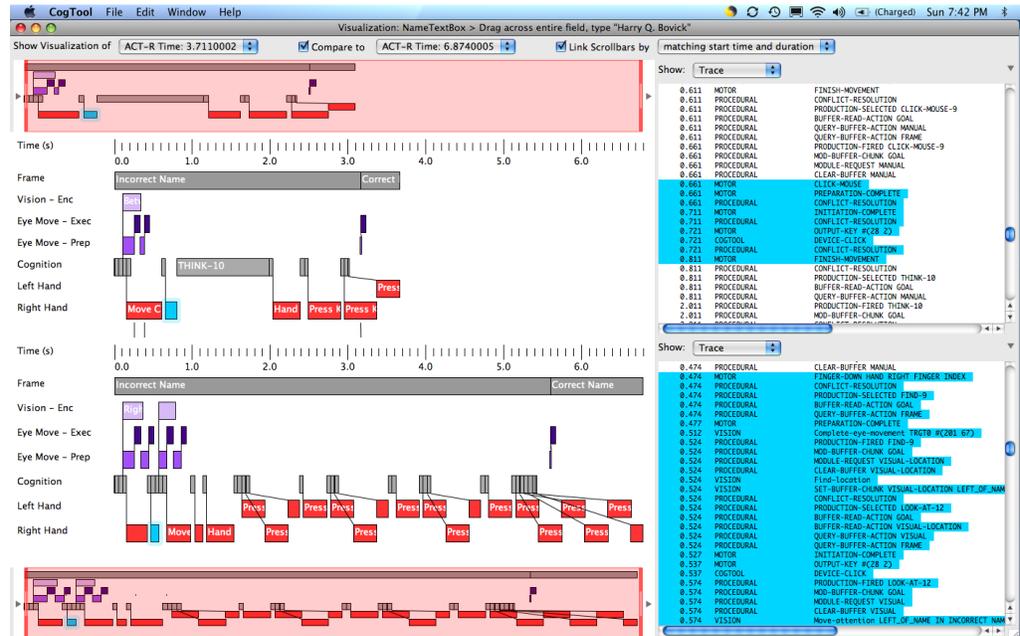


Figure 5-11: Comparison of two methods for editing in the ACT-R visualization windows.

REFERENCES

- Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.
- Card, S.K., Moran, T.P., and Newell, A. (1980) The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communications of the ACM*, 23(7), 1980, 396-410.
- Card, S. K., Moran, T.P. and Newell, A. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA (1983).
- John, B. E., Prevas, K., Salvucci, D. D., Koedinger, K. (2004) Predictive human performance modeling made easy. *Proceedings of CHI 2004* (Vienna, Austria, April 2004) ACM New York. 455-462.
- John, B. E. & Salvucci, D. D. (2005) Multi-Purpose Prototypes for Assessing User Interfaces in Pervasive Computing Systems. *IEEE Pervasive Computing* 4(4), 27-34.
- John, B. E., Salvucci, D. D., Centgraf, P., Prevas, K., (2004) Integrating models and tools in the context of driving and in-vehicle devices. *Proceedings of International Conference on Cognitive Modeling* (Pittsburgh, PA, July 30 – August 1, 2004).
- Landay, J. A., & Meyers, B. A. (1995). Interactive Sketching for the Early Stages of User Interface Design. *Proceedings of CHI 1995*, (Denver, Colorado, May 7-11 1995) ACM, New York.
- Lane, D. M., Napier, H. A., Batsell, R. R. & Naman, J. (1993) Predicting the skilled use of hierarchical menus with the keystroke-level model. *Human-Computer Interaction.*, 8, 185-192
- Lin, J., Newman, M. W., Hong, J. I., and Landay, J. A. DENIM: Finding a tighter fit between tools and practice for web site design. CHI 2000, ACM Conference on Human Factors in Computing Systems, *CHI Letters*, 2(1), pp. 510-517.
- Luo, L. & John, B. E. (2005) Predicting task execution time on handheld devices using the Keystroke-Level Model. *Extended Abstracts of CHI 2005* (Portland, Oregon, April 2-7, 2005) ACM, New York.
- Luo, L. & Siewiorek, D. P. (2007) KLEM: A Method for Predicting User Interaction Time and System Energy Consumption during Application Design. *Proceedings of the 11th Annual IEEE International Symposium on Wearable Computers*. (Boston, MA, October 11-13, 2007) IEEE., Los Alamitos, CA
- Nielsen, J., and Phillips, V. A. (1993) Estimating the relative usability of two interfaces: heuristic, formal, and empirical methods compared. *Proceedings of CHI 1993*, ACM New York. 214-221.

- Teo, L. & John, B. E. (2008a) CogTool-Explorer: Towards a Tool for Predicting User Interaction. *Extended Abstracts of CHI 2008* (Florence, Italy, April 5 – 10, 2008) ACM, New York.
- Teo, L. & John, B. E. (2008b) Towards Predicting User Interaction with CogTool-Explorer. *Proceedings of the Human Factors and Ergonomics Society 52nd Annual Meeting* (New York City, New York, Sept 22-26, 2008).
- Teo, L., John, B. E., & Pirolli, P. (2007) Towards a tool for predicting user exploration. *Extended Abstracts of CHI 2007* (San Jose, USA, April 28 – May 3, 2007) ACM, New York.

APPENDIX A: DEVICES

You must choose at least one input and one output device for your Design.

Input Devices

Keyboard

A keyboard is a regular QWERTY keyboard that sits on a desk or lap, on which a person types with two hands. Use the keyboard device as an approximation to any QWERTY keyboard, e.g., full-sized hard-wired or wireless, laptop, external keyboard for a hand-held device, etc.

On screen keyboards, such as those that display on a PDA are not keyboard devices but touchscreen devices.

CogTool’s predictions of human performance assume an average typing speed of 41 word per minute for this device.

When a keyboard is include as an input Device, an area labeled “Keyboard” appears on the bottom of each Frame so you can define Transitions from the keyboard.

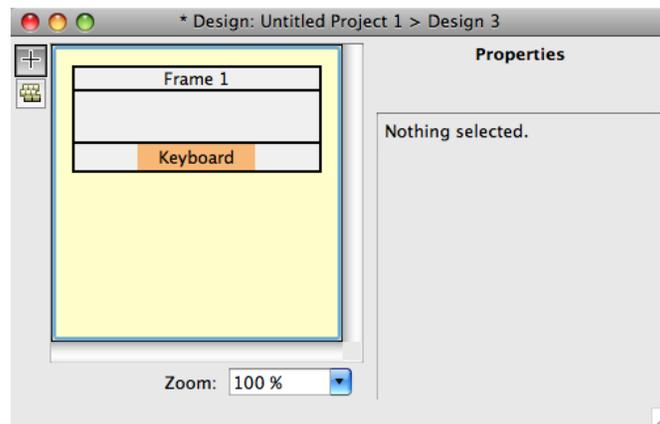


Figure A-1: Example of a Frame shown in its Design widow that has only a keyboard Device.

Mouse

A mouse is a regular computer mouse. That is, a pointing device that fits in the hand and is moved on a flat surface to position a cursor on the display.

CogTool’s predictions of human performance use Fitts’s Law to estimate mouse movement time and add additional time for the click when the mouse reaches its destination. CogTool’s mouse has three buttons (left, middle, right).

When the mouse is included as an input Device, the Design can include widgets that require a pointing device, e.g., buttons, menus, check boxes, etc. The Transitions can include mouse transitions, e.g., click, double-click, shift-click, etc.



There is currently no way to represent the IBM TrackPoint, a track ball, or a track pad in CogTool.

Touchscreen

A touchscreen is a display that reacts to touch. The touch can be with a finger or a stylus, but it doesn't require a click as a mouse does. Use the touchscreen Device to approximate any device that uses the finger to operate, even if it has hardware buttons instead of a touch screen. Thus, touchscreen is the device to select for prototyping PDA, cell phone, in-vehicle devices like GPS systems, etc.

CogTool's predictions of human performance use Fitts's Law to estimate movement time but since Fitts's Law was originally determined using tapping with a stylus, there is no additional time added for the tap.

When the touchscreen is included as an input Device, the Design can include widgets that require a pointing device, e.g., buttons, menus, check boxes, etc. In addition, a Graffiti Widget becomes available. The Transitions can include touchscreen transitions, e.g., tap, double-tap, Graffiti strokes, etc.

Microphone

A microphone inputs sound into the system so it can react to that sound. Use a microphone for any system with speech input or any environment where you want to simulate users speaking to each other. For example, include a microphone if you are prototyping systems in an airplane cockpit because the pilot often talks with the Air Traffic Controller.

CogTool's predictions of human performance use ACT-R's speaking model. That model uses 50 ms per character as an estimate for how long it takes the user to say words into the microphone.

When a microphone is included, an area labeled "Microphone" appears on the bottom of each Frame so you can define Transitions from the microphone.

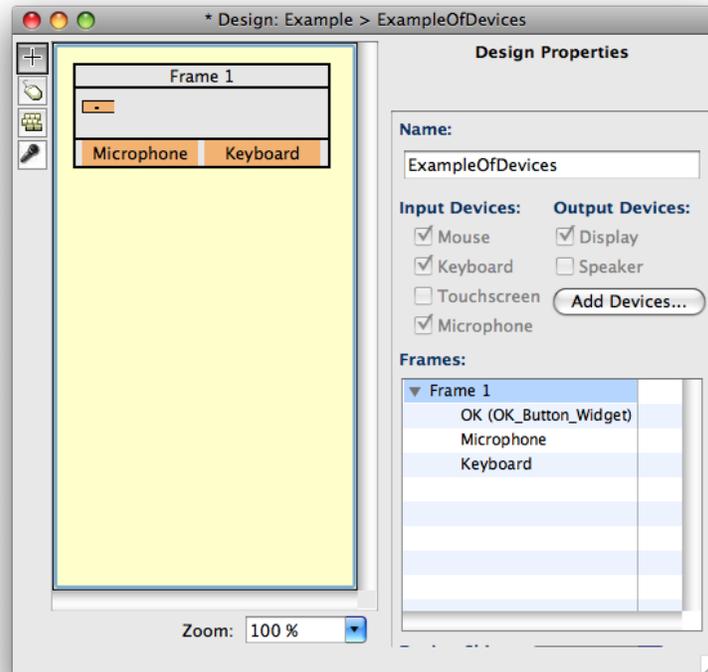


Figure A-2: Example of a Frame shown in its Design widow that has a keyboard, a mouse and a microphone.

Output Devices

Display

A display is any device that presents information visually to the user. CogTool assumes that every Design has a display Device, thus, the display device is selected by default and cannot be deselected.

The display contains all the widgets you put in the Frames.

Speaker

A speaker presents auditory sounds or words to the user. Use a speaker whenever the user gets information from the Design through sound, or any environment where you want to simulate users speaking to each other. For example, include a speaker if you are prototyping systems in an airplane cockpit because the pilot will receive both verbal communications from Air Traffic Control and auditory signals from the cockpit systems (e.g., alarms and alerts) while flying.

CogTool's predictions of human performance use ACT-R's hearing model. That model uses 50 ms per character as an estimate for how long it takes the user to hear and comprehend words emitted by the speaker.

When a speaker is included as an output Device, a text field appears at the top of each Frame in the Design where you can type in the sounds or words the speaker will emit when this Frame is entered.

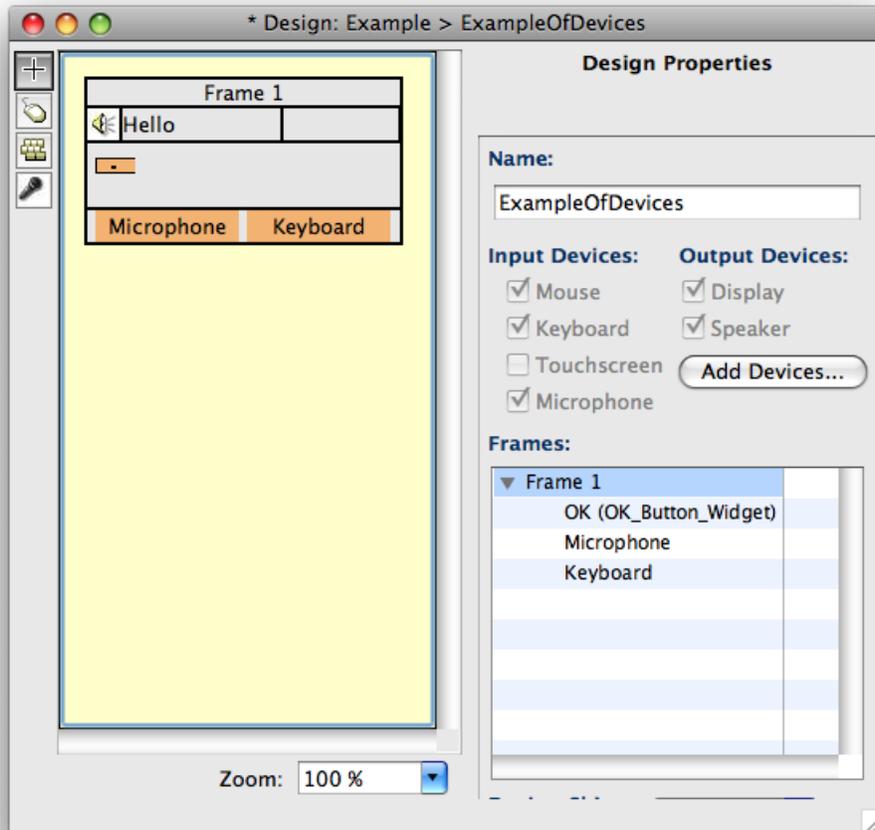


Figure A-3: Example of a Frame shown in its Design window that has a speaker, as well as the keyboard, mouse and microphone. The speaker will emit the word “Hello” on entering this Frame..

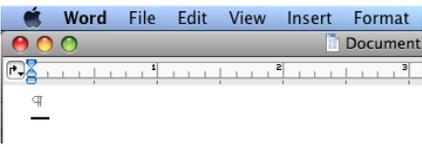
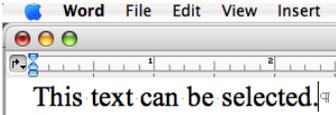
APPENDIX B: TYPES OF WIDGETS AND WHEN TO USE THEM

Widget	Examples	Use this Widget...
<p>Button</p> 		<p>...anytime a Widget looks like a button as defined by current conventions. The button can be a hardware button (like the Palm® Datebook button to the left), a static silk-screened button (like the Palm® calculator button), or a button that appears dynamically on a screen (like the New and Sign In buttons). A button Widget is also used for things like the Close, Hide, and Full Size buttons that control Mac windows; the back, forward, reload, stop, and home buttons on the Firefox toolbar; and tabs in Firefox (two of which are shown). The Close button within each Firefox tab would also be represented by a button Widget.</p> <p>Standard button widgets can be configured to appear to toggle from unpressed to pressed when you demonstrate tasks on them. If you want that behavior with Custom button widgets, you have to use two Frames with two images for the buttons, with transitions between them. Using Standard button widgets saves a lot of work.</p>
		<p style="text-align: right;"><i>More Widgets on next page . . .</i></p>

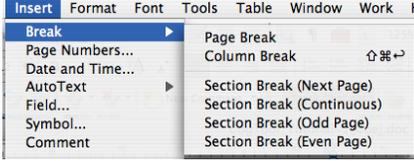
Widget	Examples	Use this Widget...
<p>Link</p> 	<p>http://www.cogtool.org</p> <p>Carnegie Mellon University</p> <ul style="list-style-type: none"> ● About Us Contact Info Facilities Visit HCII <p>What's New</p>  <p>Available for order: Updated 2006 Research-Based Web Design & Usability Guidelines Book</p>	<p>...anytime a Widget represents a link on a web page. This can be a URL, a text link, or anything else that acts as a link in a web page or document (top three pictures). For example, in the bottom picture, both the blue words and the picture of the book on the “What’s New” section of Usability.gov are links.</p> <p>Standard link widgets and Custom link widgets are the same.</p>
		<p style="text-align: right;"><i>More Widgets on next page . . .</i></p>

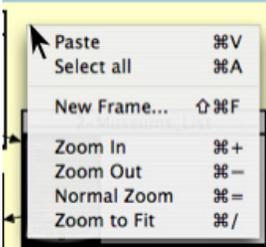
Widget	Examples	Use this Widget...
<p>Checkbox</p> 	<p>Private: <input type="checkbox"/></p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Allow anonymous posts. <input type="checkbox"/> Allow author to modify message after posting. <input type="checkbox"/> Allow author to remove own posted messages. <input checked="" type="checkbox"/> Allow file attachments. <input checked="" type="checkbox"/> Allow new threads. 	<p>...anytime there is an interactive box that can be toggled on or off (checked or not checked) independently of any other checkboxes. The top example is a single checkbox from a Palm® calendar entry setting. The bottom example is a set of related checkboxes encountered when setting up a discussion forum on Blackboard®.</p> <p>Standard checkbox widgets appear to toggle from unchecked to checked when you demonstrate tasks on them. If you want that behavior with Custom checkbox widgets, you have to use two Frames with different images for the checkboxes in different states, with transitions between them. Using Standard checkbox widgets saves a lot of work.</p>
		<p style="text-align: right;"><i>More Widgets on next page . . .</i></p>

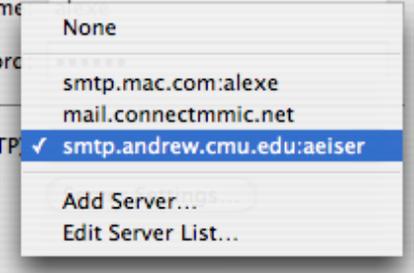
Widget	Examples	Use this Widget...
<p>Radio Button</p> 	 	<p>...anytime there is a set of visible options where (1) the options remain visible after clicking on one, (2) the state of the option is visible and remains visible, and (3) only one option can be selected at a time. These often have circles beside them (like the text style selections to the left) that show the state of the options, but they may also take other shapes (like the days of the week in the Palm® Date Book, to the left).</p> <p>Standard radio button widgets appear to toggle between unselected and selected when you demonstrate tasks on them, and they automatically enforce that only one is selected at a time. If you want that behavior with Custom radio button widgets, you have to use many Frames with many images for the radio buttons, with transitions between them, for each state the set of radio buttons can be in. In addition Standard radio button widgets automatically align as you created them and can be resized and re-aligned as a group. Using Standard radio button widgets saves a lot of work.</p>
		<p><i>More Widgets on next page . . .</i></p>

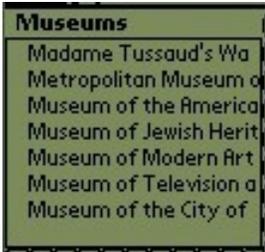
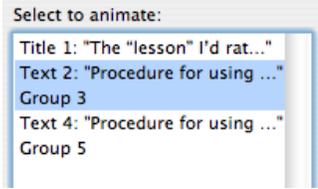
Widget	Examples	Use this Widget...
<p>Text Box</p> 	 	<p>...anytime there is a place to enter text (letters or numbers). These can have different looks, such as the Username and Password boxes on Gmail®, the text area for the number of days on the Palm®, or even the entire page area in MSword.</p> <p>NOTE: A text box is different from the text <i>inside</i> the text box. See the Section Working with Text Boxes and Text.</p>
<p>Text</p> <p>Text</p>		<p>...for any text that can be selected and copied, deleted, pasted over, or typed over. It can be the text in a text box, the text someone is editing in a word processor, text on the web you want to copy, etc.</p> <p>See the Section Working with Text Boxes and Text.</p> <p style="text-align: right;"><i>More Widgets on next page . . .</i></p>

Widget	Examples	Use this Widget...
<p>Menu</p> 		<p>...for the menu headers. In the Palm® menu to the left, “Record,” “Edit,” and “Options” would be menu Widgets. In the Microsoft Word™ menu below in the next section, “Insert,” “Format,” and all the words across the top would be menu Widgets.</p> <p>Standard menu widgets allow you to type in entire hierarchical menu systems with menu headers, items and as many levels of submenus as your prototype needs. These menus automatically expand and contract as you demonstrate tasks on them. If want this behavior with Custom menus, you would have to use menu widgets, submenu widgets and menu item widgets separately. This is not only much more effort to construct, with many more Frames, but it has proven to be error-prone. We strongly recommend that you use Standard menu widgets unless your menu system does not look like a standard hierarchical menu system, e.g., you are using pie menus.</p>
		<p style="text-align: right;"><i>More Widgets on next page . . .</i></p>

Widget	Examples	Use this Widget...
<p>Submenu (Necessary in Custom only)</p> 		<p>...for menu items that produce another menu. The Palm® menu in the previous section does not have a submenu Widget. The Microsoft Word™ menu on the left would have two submenu Widgets, “Break” and “AutoText.”</p> <p>Standard menu widgets remove the necessity to use submenu widgets and menu item widgets. We strongly recommend that you use Standard menu widgets.</p>
<p>Menu Item (Necessary in Custom only)</p> 		<p>...for items that can be selected from a menu (and do not produce other menus). In the Palm® menu to the left, “Undo,” “Cut,” and all the words below them would be menu item Widgets. In the Microsoft Word™ menu in the previous section, “Page Break,” “Column Break,” and all the words below them would be menu items.</p> <p>Standard menu widgets remove the necessity to use submenu widgets and menu item widgets. We strongly recommend that you use Standard menu widgets.</p>
 <p>Using CUSTOM Menu, Submenu, and Menu Item: When demonstrating a Task, you can only interact with a submenu item Widget or a menu item Widget after having interacted with its associated menu Widget. This is true because a user cannot interact with a submenu or menu item until they are visible, and they are not visible until the menu is pulled down. If you interact with a submenu item or a menu item Widget without first interacting with its associated menu Widget, CogTool will give you a warning message.</p>		
		<p><i>More Widgets on next page . . .</i></p>

Widget	Examples	Use this Widget...
<p>Context Menu</p> 		<p>...for a menu that is accessible by right-click (PC) or CTRL-click (Mac).</p> <p>Standard context menu widgets allow you to type in entire hierarchical menu systems with as many levels of submenus as your prototype needs. These menus automatically expand and contract as you demonstrate tasks on them. If you want this behavior with Custom context menus, you would have to use context menu widgets, submenu widgets and menu item widgets separately. This is not only much more effort to construct, with many more Frames, but it has proven to be error-prone. We strongly recommend that you use Standard context menu widgets unless your menu system does not look like a standard hierarchical menu system, e.g., you are using pie menus.</p>
		<p style="text-align: right;"><i>More Widgets on next page . . .</i></p>

Widget	Examples	Use this Widget...
<p>Pull-Down List</p> 	  	<p>...anytime a list drops down in response to the click (or tap) on the displayed item, only one item on the list can be selected, and the selected item replaces the original displayed item as the label of the list. Use the pull-down list Widget for the displayed item (before the list is pulled down). An example is the list of types of “To Do” items to display on the Palm®. In the example to the left, the “All” with the down arrow indicates a pull-down list; if this were the image in the Frame, the arrow+All would be a pull-down list Widget. In the example from Apple’s Mail, the object with the server_name+arrows would be a pull-down list Widget.</p> <p>Standard pull-down list widgets allow you to type in all the items in the list and identify which item should be initially selected. These lists automatically expand and contract as you demonstrate tasks on them. If want this behavior with Custom pull-down lists, you would have to use pull-down list widgets and pull-down list items separately. This is not only much more effort to construct, with many more Frames, but it has proven to be error-prone. We strongly recommend that you use Standard pull-down list widgets.</p>
		<p><i>More Widgets on next page . . .</i></p>

Widget	Examples	Use this Widget...
<p>Pull-Down Item (Necessary in Custom Only)</p> 		<p>...for the items in the pull-down list that could be selected after the list is pulled down. In the example on the left from Palm®, if this were the image in the Frame, each of the items would be pull-down item Widgets. Likewise, all the items in Apple's Mail pull-down would be pull-down item Widget.</p> <p>Standard pull-down list widgets remove the necessity to use pull-down list item widgets. We strongly recommend that you use Standard pull-down list widgets.</p>
 <p>Using CUSTOM Pull-Down Lists and Items: When demonstrating a Task, you can only interact with a pull-down item Widget after having pulled down the list by interacting with its associated pull-down list Widget. This is true because a user cannot interact with list items until they are visible, and they are not visible until the list is pulled down. If you demonstrate a Task that interacts with a pull-down item Widget without first interacting with the associated pull-down list Widget, CogTool will give you a warning message.</p>		
<p>List Box Item</p> 	 	<p>...when there is a visible list of items, from which one or more can be selected, that does not have to be pulled down or popped up. The Museums list to the left is from the ChoiceWay™ Guide to New York City; the animation list box is from the “Custom Animation” dialog in PowerPoint™.</p> <p>Standard list box item widgets and Custom list box item widgets are the same.</p>
<p><i>More Widgets on next page . . .</i></p>		

Widget	Examples	Use this Widget...
<p>Graffiti®</p> 		<p>...to designate an area on a touchscreen that can interpret Graffiti® strokes. To mock-up a Palm® interface, we suggest using one Graffiti® Widget on the left side called “Graffiti® Letters” and one on the right side called “Graffiti® Numbers” because the user must target one or the other side when using a Palm®. Other areas of the touchscreen can also receive Graffiti® strokes and can be designated with Graffiti® Widgets.</p> <p>Standard Graffiti® widgets and Custom Graffiti® widgets are the same.</p>
<p>Non-Interactive</p> 		<p>...where a user might have to look to get information but cannot manipulate that information from the Widget. The example to the left is a picture of part of the Flight Management System of a Boeing 777 airplane. The Altitude display (currently displaying the number 10,000) would be defined as a non-interactive Widget. The Task could involve looking at that Widget, but the display cannot be changed directly (the knob below it would have to be defined as an interactive Widget and separately manipulated).</p> <p>Standard non-interactive widgets and Custom non-interactive widgets are the same.</p> <p style="text-align: right;"><i>End of Widgets</i></p>

APPENDIX C: STEPS PLACED AUTOMATICALLY BY COGTOOL

As you demonstrate tasks in CogTool, it automatically places steps that are associated with your demonstrated actions and that create a valid cognitive model from your demonstration. These steps appear in yellow in the Script Step window (your demonstrated steps appear in white).

Specifically where a step is placed automatically is a complicated algorithm based on transforming the rules written by Card, Moran and Newell (1980, CMN) into rules associated with widgets. We cannot explain all the reasoning in this appendix because it requires a deep understanding of CMN's rules, the notion of a “cognitive unit” and other concepts in the 1980 paper and subsequent book (Card, Moran and Newell, 1983) and we direct you to those sources for additional information. The ultimate definition of where these steps are placed is the CogTool code (which you can get under an Open Source license from the CogTool Project), but below is a summary of how these steps are placed.

A **Home** step is placed when

- the hands are on the keyboard and the next action is with the mouse, because the hand is needed on the device to perform actions with that device.
- the hand is on the mouse and the next action uses that hand to type on the keyboard.



CogTool takes into account whether the mouse is used with the right hand or the left (a setting available in the Script window) and whether typing will need to use the hand the mouse is on.

A **Think** step is placed when

- you explicitly insert a Think step into the demonstration with the “Think” button.
- the previous step is not a Think step (because the KLM does not allow more than one Think in a row)
- * you explicitly insert a Look-At step into the demonstration with the Look-At button, because CogTool assumes that the reason for deliberately looking at a widget is to read and comprehend the information in that widget and this comprehension time is represented as a Think.. The Think step is inserted after the Look-At step.
- * a step is demonstrated on the keyboard, Graffiti®, or microphone that you labeled as a “command” (by checking the “command” checkbox in the Transition Properties pane) because the CMN's rules inserted mental operators before “commands, not arguments”. The Think step is inserted before the demonstrated step.
- * a step is demonstrated on the keyboard that is the typing the first command key (e.g., control, escape, etc.) in a string of command keys, because command keys are usually associated with commands to the system, not arguments. But a string of command keys are usually what CMN referred to as a “cognitive unit” so no Think steps go between them by CMN's rules. The Think step is inserted before the demonstrated step.
- * a custom Menu Item is the target of a hover or click/tap action, because picking a menu item is usually a command, not an argument and Lane et. al. (1993) presents data that suggest that the only

pause occurs before the selection of the menu header. The Think step is inserted before the step that targets the item's Menu Header.

- * a standard Menu Item is the target of a hover or click/tap action. Because of Lane et.al.'s data, CogTool places the Think step before the step that targets the item's Menu Header.
- * a step is demonstrated on a context menu item widget, dropped down by either a right single click (typical for a system with a two-button mouse) or a ctrl-left single click (typical for a Mac with one mouse button). Although the target may be either a command or an argument, the act of bringing up a context menu signals the intention to invoke a command and the selected item in the menu is typically a command. The Think step is inserted before the mouse movement that selects the item.
- * a click/tap action is demonstrated on a Text Box that is not "identical" to the last step's target, because selecting a Text Box is a command to set the focus in that box. Contrast this to clicking on a text widget, which selects an argument (the text), which does not get a Think step. The Think step is inserted before the demonstrated step.
- * a click/tap action is demonstrated on a Button/Link that is not "identical" to the last step's target AND the two widgets in question are not both "OK" terminators nor both "CANCEL" terminators. Buttons and links are usually commands to the system, but if the user has to click twice on the same button or link, that is probably a "cognitive unit" so there should be no Think step between click on identical buttons/links (e.g., clicking the back button repeatedly in a web browser to get back a few pages). The restriction about terminating buttons reflects CMN's rules about redundant terminators. The Think step is inserted before the demonstrated step.

Think steps are not placed before demonstrations on Checkbox widgets Radio Button widgets, List Box widgets, Pull-down List items, or Text widgets because these widgets typically select arguments, not commands.

As said above, this is only a summary of the rules and the CogTool code may contain subtleties more easily expressed in code than in prose.

APPENDIX D: SPECIAL KEYS AND SYMBOLS

A **special key** is a key that performs a special function (as opposed to standard letter and number keys). CogTool uses symbols to represent these keys, which you encounter anytime you use them in a keyboard Transition. If you export your CogTool Scripts to CSV and open them in other applications (e.g. Microsoft Excel), these applications may not be able to display the symbols and must therefore use substitutes. Listed below are all fifteen special keys, the symbols used to represent them in both the Windows and Macintosh versions of CogTool, and the symbols' corresponding substitutes when they cannot be displayed.

Special Key	Win Symbol	Mac Symbol	Substitute for Symbol
Shift	⇧	⇧	(SHIFT)
Control	⌘	⌘	(CTRL)
Alt	⌥	⌥	(ALT)
Command	⌘	⌘	(CMD)
Function	ƒ	ƒ	(FN)
Enter	CR	↵	(ENTER)
Escape	ESC	⌫	(ESC)
Tab	→	⇒	(TAB)
Backspace	BS	⇐	(BS)
Delete	DEL	⌫	(DEL)
Caps Lock	⇧	⇧	(CAPS)
Up	↑	↑	(UP)
Down	↓	↓	(DOWN)
Left	←	←	(LEFT)
Right	→	→	(RIGHT)

APPENDIX E: REPORTING A BUG

Though fully functional, CogTool is still a work in progress and will sometimes crash. A CogTool crash generates an error window like the one seen below.

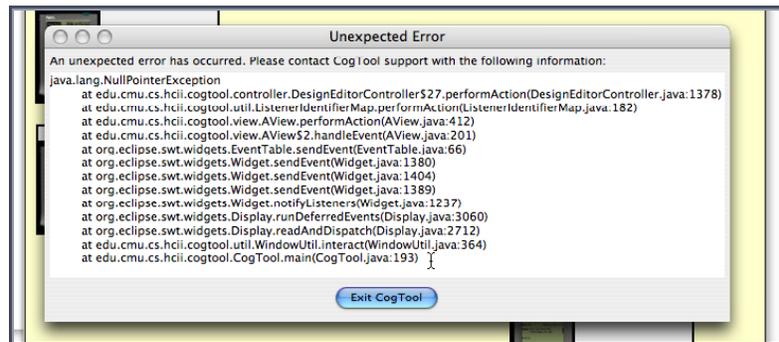


Figure A-4: Error message window

CogTool crashes result from bugs in the program that have not yet been discovered or fixed. Reporting these bugs alerts us to problems we can address for future releases of CogTool.

To report a bug:

- Copy the text in the error window by highlighting the text and copying with CTRL+C (Windows) or ⌘C (Mac OS).
- Identify the version of CogTool you have.
Mac OS X: Under the CogTool menu listing, choose About CogTool.
Windows: Under the Help menu listing, choose About CogTool.



5. Send an email to cogtool@cs.cmu.edu with the text from the error message, the version of CogTool you are using, and a description of the action you performed immediately before the error appeared. If we need additional information, we will contact you. **Note:** If you have a .cgt file and are willing to share it, please send the file as an attachment.

We will do our best to resolve the problem and will contact you with a fix or workaround as soon as one is available.

APPENDIX F: COGTOOL SHORTCUT KEYS

The following shortcut keys can be used whenever the action they perform is required.

Save	Save As	Open Project	New Project	Close Window
Mac: ⌘S	Mac: ⇧⌘S	Mac: ⌘O	Mac: ⌘N	Mac: ⌘W
Win: CTRL+S	Win: CTRL+SHIFT+S	Win: CTRL+O	Win: CTRL+N	Win: CTRL+W

Cut	Copy	Paste	Duplicate	
Mac: ⌘X	Mac: ⌘C	Mac: ⌘V	Mac: ⌘D	
Win: CTRL+ X	Win: CTRL+C	Win: CTRL+V	Win: CTRL+D	

Delete	Rename	Edit	Undo	Redo
Mac: ⌘⌫	Mac: ⌘R	Mac: ⌘E	Mac: ⌘Z	Mac: ⌘Y
Win: CTRL+DELETE	Win: CTRL+R	Win: CTRL+E	Win: CTRL+Z	Win: CTRL+Y

The remaining shortcut keys perform actions that are specific to the CogTool window you are using.

Design
New Design
Mac: ⇧⌘D
Win: CTRL+SHIFT+D

Frame		
New Frame	Setting Background Image	Removing Background Image
Mac: ⇧⌘F	Mac: ⌘B	Mac: ⇧⌘B
Win: CTRL+SHIFT+F	Win: CTRL+B	Win: CTRL+SHIFT+B

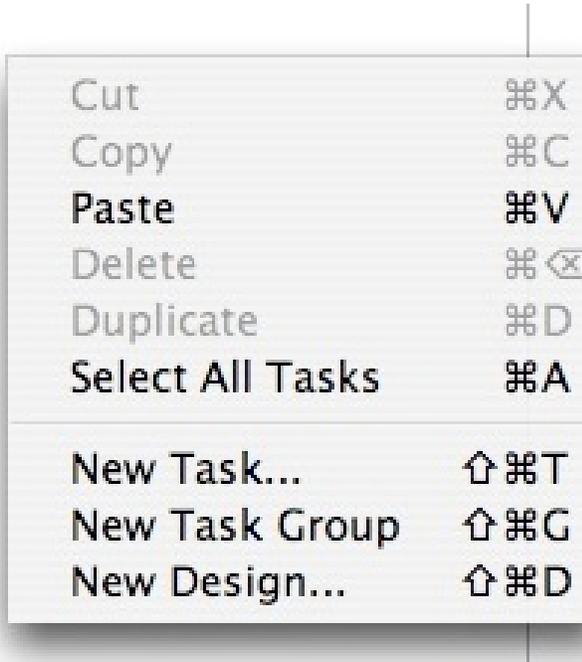
Widget	
New Widget	Setting Layer Color
Mac: ⇧⌘I	Mac: ⇧⌘C
Win: CTRL+SHIFT+I	Win: CTRL+SHIFT+C

Task/Task Group	
New Task	New Task Group
Mac: ⌘T	Mac: ⌘G
Win: CTRL+SHIFT+T	Win: CTRL+SHIFT+G

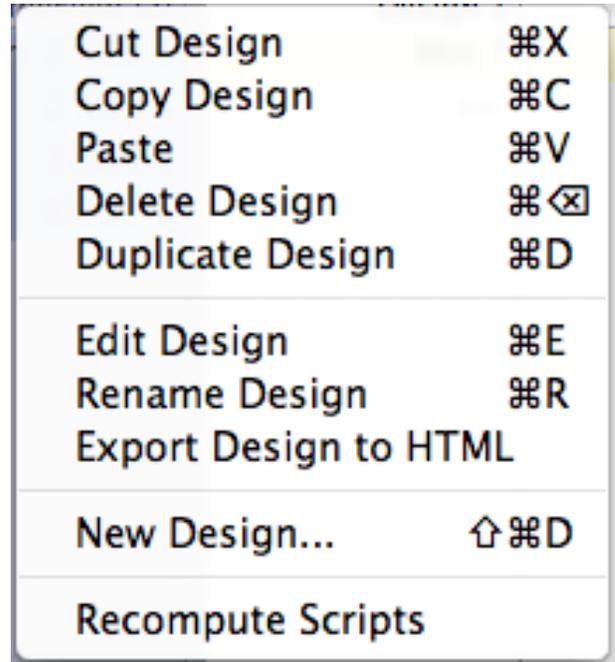
Zoom for Design, Frame, Script			
Zoom In	Zoom Out	Normal Zoom	Zoom to Fit
Mac: ⌘+	Mac: ⌘-	Mac: ⌘0	Mac: ⌘/
Win: CTRL+=	Win: CTRL+ -	Win: CTRL+0	Win: CTRL+ /

APPENDIX G: COGTOOL CONTEXT MENUS

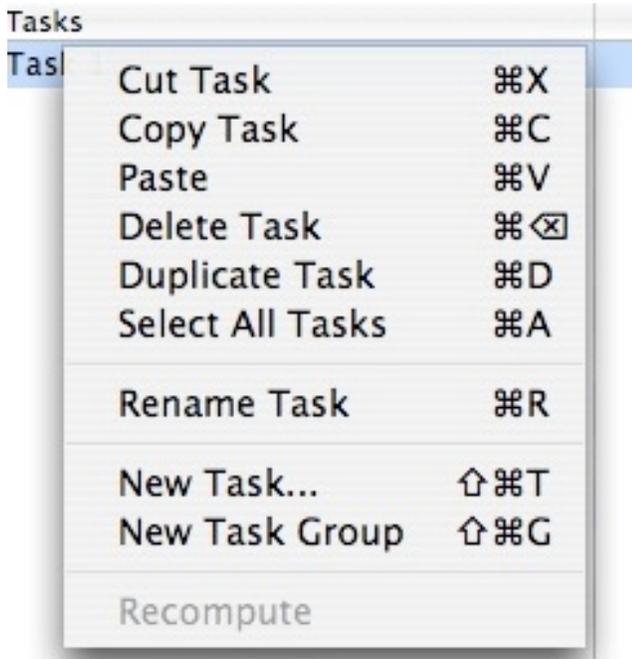
Project Window



Context Menu - Nothing selected

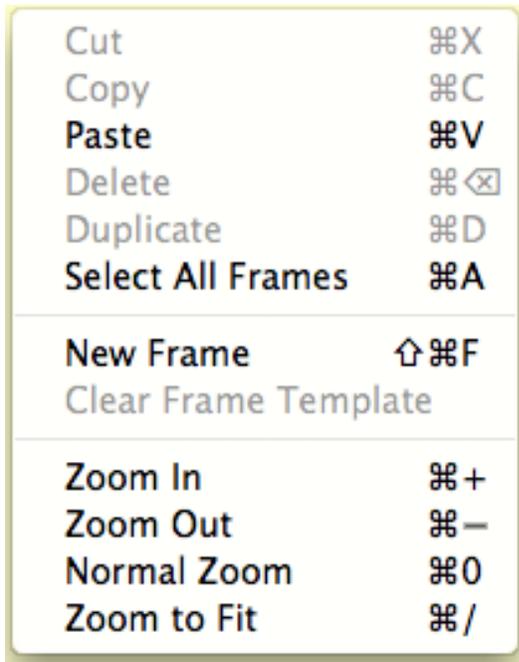


Context Menu - Design selected

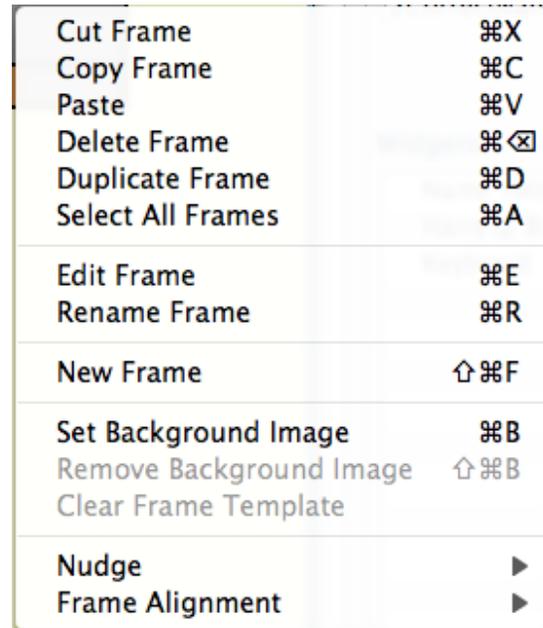


Context Menu - Task selected

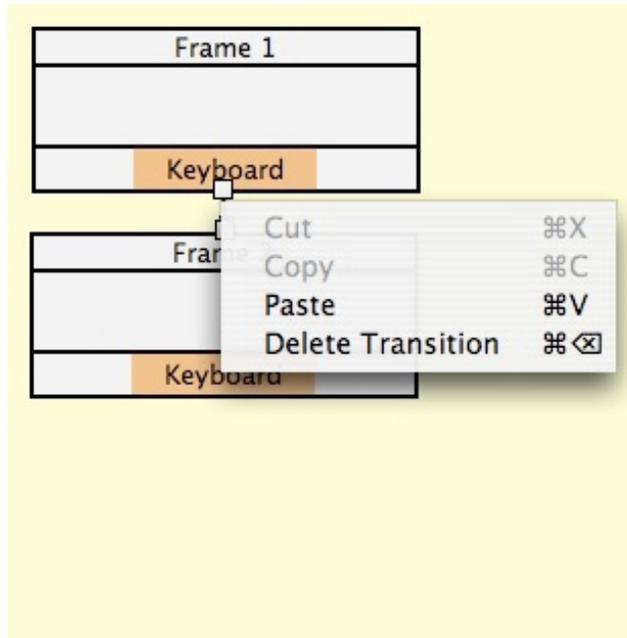
Design Window



Context Menu - Nothing selected

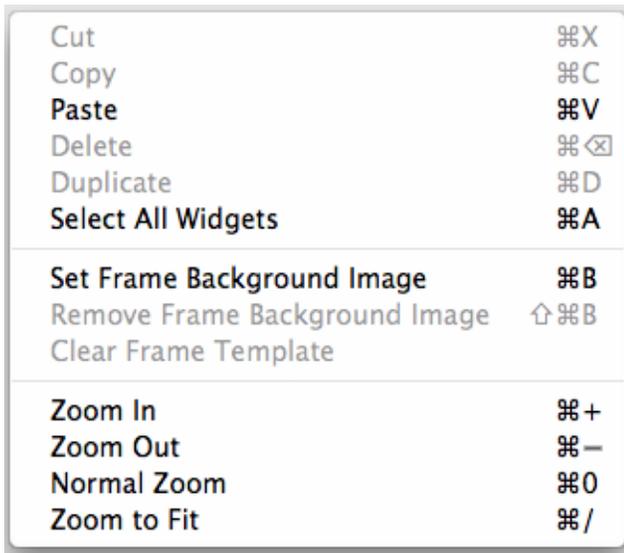


Context Menu - Frame selected

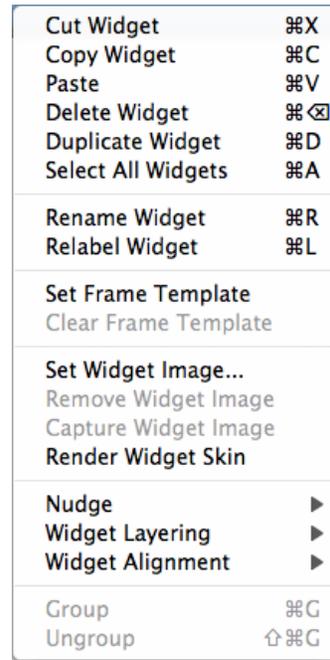


Context Menu - Transition selected

Frame Window

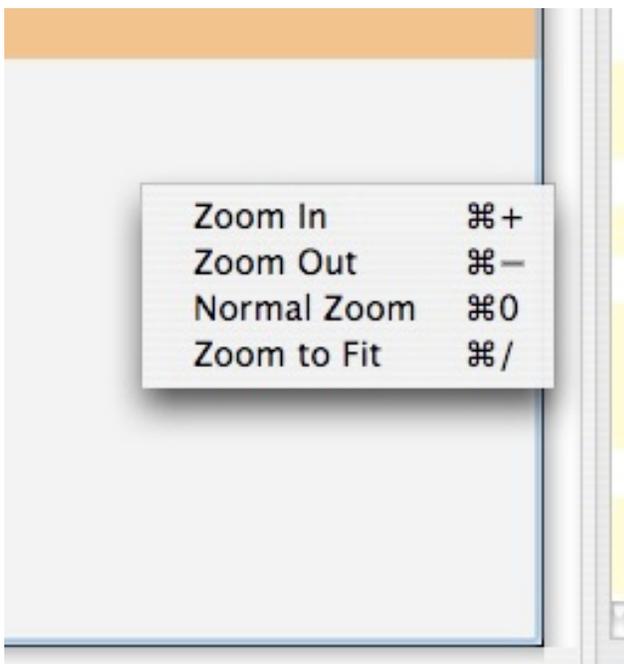


Context Menu - Nothing selected

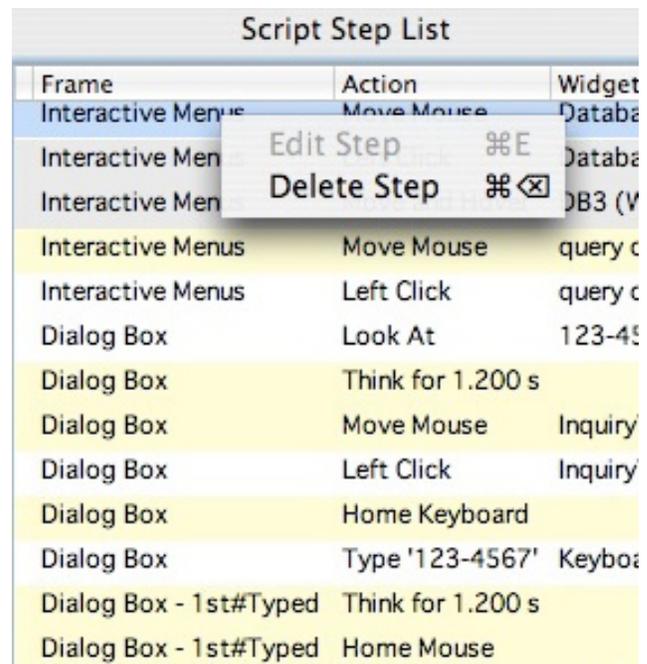


Context Menu - Widget selected

Script Editor



Context Menu - Left pane



Context Menu - Step list

APPENDIX H: GLOSSARY

ACT-R

ACT-R is a cognitive theory of how human perception, cognition, and motor actions. It is a computational theory that is implemented in Lisp and runs inside CogTool to produce its predictions of human performance. For more information on ACT-R, go to : <http://act-r.psy.cmu.edu/about/>.

Demonstration

A Demonstration follows a series of Transitions from a start Frame to the end Frame of a Task. CogTool produces a Script from a Demonstration and then computes a predicted execution time for the Script using ACT-R.

Design

A Design is a series of Frames that represent the user interface of the system being analyzed and the Transitions that move from Frame to Frame.

Frame

A Frame represents a single screen of a user interface. Multiple Frames can be linked together with Transitions to form a complete Design.

Keystroke-Level Model (KLM)

Keystroke-level model (KLM) is one of the frameworks used by CogTool to make predictions on how long it would take to perform a Task using a keyboard and mouse. For more information on the KLM, see Card, Moran and Newell, 1983.

Project

A CogTool Project stores all the interface Designs, Tasks that can be performed on the Designs, and predictions of user behavior. A Project is stored in a .cgt file.

Script

A Script is a list of steps necessary to accomplish a Task. A Script is formed from a Demonstration and has added steps in accordance with the Keystroke-Level Model.

Task

A Task represents a specific sequence of actions a user will take to accomplish a goal.

Task Group

A Task Group organizes the Tasks and displays the sum, mean, minimum, or maximum skilled execution time predicted for the Tasks in that group.

Transition

A Transition represents the user's action that causes the system to move from one Frame to the next. A Transition goes from a Widget to a Device to a Frame.

Widgets

A Widget is an element on a Frame with which a user may interact. A Widget could be a button, link, checkbox, radio button, text box, text, menu, submenu, menu item, context menu, pull-down list, pull-down item, list box item, or Graffiti®. In CogTool, Widgets can also be created for non-interactive elements to construct a full representation of an interface.