F Usability Findings

This appendix presents the results of the usability study. The findings for which I present concrete recommendations are organized according to *scope* and *severity* [10], and presented in the first four sections of this appendix. I begin this appendix with a brief overview of scope and severity for those unfamiliar with the terms.

The scope of a problem indicates how widespread that problem is. A problem with *local* scope is limited to one or two screens while a problem with *global* scope applies to multiple screens. A problem with global scope is more critical than one with local scope.

The severity of a problem indicates how critical the problem is. Severity is measured on a scale of 1 to 4, with 1 being the most severe. The four levels are described by Dumas and Redish and summarized below:

> *Level 1*: prevent completion of a task
>
> *Level 2*: create significant delay and frustration
>
> *Level 3*: have minor effects on usability
>
> *Level 4*: represent subtle problems and may point to potential enhancements

Each finding described in this appendix is supported by some sort of evidence. I use five forms of evidence:

> *Heuristic Violation:* the interface violates a known heuristic[4]
>
> *Tutorial Result:* users experienced a problem during the Tutorial
>
> *Exercise Result:* users experienced a problem during the Exercises
>
> *Qualitative Result:* users commented in the Evaluation Questionnaire
>
> *Comment IDs:* users commented in their verbal protocol

Heuristic violations refer to places where the interface violates one of the heuristics described in Section 3.3 of the thesis. The remaining pieces of evidence are identified by their identification numbers and described in Appendix E.

One or more forms of evidence support each finding documented in this appendix. In general, each finding is supported by two different forms of evidence or by three or more users. Any piece of evidence deemed particularly important is also documented. Those findings with only minimal objective evidence, but supported subjectively, are identified with a ◆ in the left margin.

[4] The identification of heuristic violations was made in retrospect. A heuristic analysis may not identify the same problems.

F.1 Level 1

A *level 1* usability problem prevents users from completing a task.

F.1.1 Global

Finding #1: Users confused by event notification
Evidence:
- **Heuristic Violation:** NA
- **Tutorial Results:** 1, 9
- **Exercise Results:** 10, 11, 12, 13
- **Qualitative Results:** NA
- **Comment IDs:** 207, 253, 258, 259, 261, 263, 267

Explanation: At various times, users became confused about the cause of an event notification. Users simply did not understand what had happened or did not associate beeping with an event notification. During the Exercises, users occasionally required more than two (and sometimes more than three) times par to report certain events.
Recommendations: Although users were not frequently confused, the fact that they were confused is troublesome. User confusion was not limited to the duration of the Tutorial so this is not simply a learning curve issue. It is important to note that user inefficiency in reporting certain event notifications could be due to poor timing during that phase of the Exercises. However, these symptoms suggest the need for a mechanism through which users can be given a written explanation of the most recent event, a sort of log or history of events. The Control Panel seems to be the obvious location for such a listing.

F.1.2 Local

Finding #2: Some help windows aren't helpful.[5]
Evidence:
- **Heuristic Violation:** Help
- **Tutorial Results:** NA
- **Exercise Results:** NA
- **Qualitative Results:** NA
- **Comment IDs:** 80, 95, 249, 251

Explanation: The help window (and informational text on the Event Configuration tab of the Control Panel) does not explain read disconnected mode. The help window associated with the Space Information indicator does not update if the space status changes while it is visible. The help window associated with the progress view of the Hoard Walk indicator was never written. The help window associated with the Event Configuration tab of the Control Panel was never updated to reflect the revised layout.
Recommendations: These help windows need to be written (or rewritten) to be consistent with the windows they describe. The Space help window needs to update when the space status changes.

[5] This finding is categorized as local because it represents four individual findings each on a single window rather than a finding about all help windows in the interface.

Finding #3: Users fail to include needed programs in definition.
Evidence:

Heuristic Violation: Memory Overload
Tutorial Results: NA
Exercise Results: 3
Qualitative Results: NA
Comment IDs: 185

Explanation: Users failed to realize that one of their task definitions required the use of the `uncompress` program because some of the data included in that definition was compressed. Two users failed to hoard the editing task as part of one of their task definitions or even as a top-level task.
Recommendations: The interface could track what programs access data contained in task definitions and then give users hints regarding any programs omitted from a task definition.

Finding #4: Event Configuration Tab of Control Panel layout inappropriate to task.
Evidence:

Heuristic Violation: Simple & Natural Dialogue
Tutorial Results: 12
Exercise Results: 5
Qualitative Results: 5
Comment IDs: 17, 162, 206

Explanation: In order for users to modify the configuration of a particular event, they must first know under what indicator that event is notified. For instance, to change the configuration of the "Task Unavailable" event, users must first select the "Task" indicator from the indicator listbox of the Event Configuration tab of the Control Panel. Then, they can select the "Task Unavailable" event from the event listbox. In this example, the decision to look under the "Task" indicator is relatively straightforward. However, if the event is "Weakly Connected Cache Miss Advice", users could reasonably look under "Network", "Space", or "Advice". Many did so. One user (S) completed an exercise incorrectly because she found the "Operating Weakly Connected" event under the "Network" indicator and assumed that to be the correct event. Other users were resorted to using brute-force search.
Recommendations: The Event Configuration tab was redesigned during pilot testing. The original design listed all events under a single listbox. One of the early pilot users pointed out that the Event Configuration should include the indicator on which an event will be notified. The remaining users in the study used the redesigned Event Configuration tab. Unfortunately, the redesign introduced the dependency between the indicator and the event. By the end of the pilot testing, it was clear that this was a problem, but the experimenter was reluctant to make further changes so late in pilot testing. The Event Configuration tab needs to be redesigned to address the original suggestion while eliminating the need to know event/indicator pairings. The recommendation is to consider a design that contains a single list of events and a copy of the indicator lights.

Finding #5: Certain characters, if used in task names, can crash the interface
Evidence:

Heuristic Violation: Speak the user's language
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 89

Explanation: If the user happens to use certain characters in the names of tasks, the interface will crash when that task is hoarded. The character in question is one of the "upper-case numbers". We need to explore the full extent of the problem to determine how many characters are involved.
Recommendations: This is a bug. Any (printable) character should be valid.

◆ **Finding #6:** Password visible on the Authentication Information window.
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 30
Explanation: As the user types into entry widgets, the widget library automatically echoes the characters that are typed to the screen. Although such behavior is appropriate for most entry widgets, it is inappropriate for one whose purpose is to collect the user's password. To avoid displaying the user's password, the interface echoes the password in the same color as the background so that the user is unable to discern the characters of the password. Unfortunately, the foreground color of the text and the background color of the entry widgets were not identical and the user could, with careful examination, read the password.
Recommendations: This problem has been corrected in the current system by making the foreground and background colors identical. Users can no longer read the password echoed to the screen. Other solutions to the problem would be to echo a '*' for each character typed into the password entry widget or to disable character echoing.

Finding #7: Users do not recognize the hoard walk advice request as a request for advice
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 155
Explanation: Three pilot users failed to recognize the hoard walk advice request as a request for advice. Two of these three users were running on an earlier version of the interface in which the hoard walk advice request came in only under the Hoard Walk indicator light. The confusion was apparently caused because they found it strange that an advice request didn't come in on the Advice indicator light. The third user, however, was running on a revised version.
Recommendations: Because two of the users involved were using an earlier version of the interface and because later versions of the interface address this problem, their confusion does not apply to the current system. The fact that a user running the revised system also did not recognize the advice request is cause for concern. However, this user made it through the Tutorial in record time, clearly skipping portions of the instructions. For this reason, even though his failure to recognize the request for what it was is a cause for concern, I do not recommend any action be taken unless further symptoms arise.

Finding #8: User does not recognize that the Hoard Walk Advice offers more details
Evidence:
Heuristic Violation: Simple & Natural
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 236
Explanation: Two users apparently didn't recognize that the Hoard Walk Advice Request offers more details. One pilot user did not recognize the window as a request for advice (see **Finding #7** above) and never expanded the "All Tasks Needing Data" pseudo-task. The second user expanded the "All Tasks Needing Data" pseudo-task, but did not explore further.
Recommendations: Although the pilot user's behavior can be explained as symptomatic of the fact that he didn't recognize the window as being a request for hoard walk advice, the other user's behavior is disturbing. Inadequate training may be partially to be blame however. The Tutorial encourages users to explore the details of the hoard walk advice request, but does not actually lead them through the process step-by-step nor does it give users reasons for selecting some items to be fetched but not others.

Furthermore, the Help window does not offer suggestions for why to select some items over others. Both of these deficiencies should be addressed. Finally, the interface should show the "All Tasks Needing Data" pseudo-task in an already expanded format so that the relevant information is available immediately.

◆ **Finding #9:** Indicator changes to Normal urgency level before appropriate to do so
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 120
Explanation: The Advice indicator change to the Normal urgency level at the point at which users double-clicked on a request for advice but before they had completed the request.
Recommendations: This bug must be corrected since users may incorrectly assume that the request for advice has been completed.

Finding #10: Users unsure whether hoard walk is proceeding during request for advice
Evidence:
Heuristic Violation: Feedback
Tutorial Results: NA
Exercise Results: 1
Qualitative Results: NA
Comment IDs: 235
Explanation: Three users commented, as they were completing the hoard walk advice request, they were unsure whether or not the hoard daemon was working on the hoard walk. One user failed to complete the hoard walk advice request and thus start the actual walk. Because she was not one of the users making this comment, it is unclear whether or not her failure to complete the request was related to confusion over whether or not hoard walk was already in progress. However, it is interesting to note that her behavior was consistent with the mistaken impression that the hoard walk was progressing as the user offered advice.
Recommendations: One possible way to eliminate this confusion is to have the hoard walk advice request come in under the Advice indicator only. The Hoard Walk indicator would change to red to indicate that it is stalled while it awaits advice. This change would make the situation more clear to users.

◆ **Finding #11:** User prioritized tasks inappropriately
Evidence:
Heuristic Violation: Help
Tutorial Results: NA
Exercise Results: 3 (a subset of the users)
Qualitative Results: NA
Comment IDs: 199
Explanation: One user prioritized tasks based not upon what tasks were most important to her work, but upon what tasks required the most resources. Two users failed to include an editing task as a subtask of their primary tasks and then failed to give it a high priority.
Recommendations: Neither the Tutorial nor the Help window explains that higher priorities should be assigned to those tasks most important to the user's work. Users were forced to make this assumption. The Tutorial and the Help window should both make it clear how priorities should be chosen. Once this omission has been corrected, we must wait to see if the problem resurfaces, perhaps by observing a few new users perform selected exercises informally.

F.2 Level 2

A *level 2* usability problem causes users significant delay or frustration.

F.2.1 Global

Finding #12: Users want undo facility.
Evidence:
Heuristic Violation: Clearly marked exits
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 12
Comment IDs: 6, 92
Explanation: Users expressed desire to undo their actions. Users accidentally deleted an entire task rather than a single element of the task. Often, they had no way to recover from their misstep.
Recommendations: The interface needs to have the ability to undo actions. In the situations observed during the user study, a single-level undo would have sufficed.

Finding #13: Users find the behavior of "Save" and "Commit" confusing
Evidence:
Heuristic Violation: Feedback
Consistency (external)
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 14
Comment IDs: 32, 39, 59, 60, 154, 158, 184, 188
Explanation: The design of the interface follows suggestions for maintaining a "unified file model" [8]. In particular, the interface

- automatically saves definitions when the user closes the definition window
- does not close a window after the user saves a definition
- does not pop-up the SAVE dialog when the user closes the definition
- allows the user to abandon all changes since opening or creating the definition
- provides an optional, manual SAVE command

The suggestions above encompass just those portions of Cooper's complete set of recommendations that seemed to apply to a simple "document" like a definition. Those recommendations not implemented include the ability to snapshot, milestone, rename, place & reposition, specify the stored format, and undo specific changes. Cooper's recommendations appeal to me because they step back and look at the functionality required of an interface from the perspective of the user rather than from the perspective of the implementation of the interface.
Recommendations: The first recommendation would be to pick a model and be consistent. If the decision is to go with Cooper's unified model, then the implementation must be modified in the following ways:

- add snapshot and milestone capabilities
- remove SAVE AS capability
- replace in-place naming with in-place renaming
- add a specialized undo facility (as discussed in Finding #12 above)

The interface does not yet need the ability to place & reposition definitions or specify the stored definition format because it manipulates such a specialized and small class of documents.

The second recommendation is to add feedback regarding the save status of definitions. Note that according to Cooper's recommendations, that feedback should not be to close the window (as five of our users expected). Perhaps the emacs * would be appropriate. A meter will probably not work well because the interface saves definitions too quickly for users to notice and understand the meter before it disappears.

Finding #14: Users are frustrated by the placement of windows on the screen.
Evidence:
Heuristic Violation: Consistency (external)
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 1
Explanation: The interface pops up window in the upper left-hand corner of the screen. Six users commented that they were frustrated with this policy.
Recommendations: The responsibility for placing new windows properly belongs to the window manager. By removing the explicit placement command, windows will be placed on the screen according to the user's default settings (generally either set to force the user to select a placement or to automatically place the window at a set offset from the last window placement). This change will make the interface consistent with the user's windowing environment.

Finding #15: Interface loses definitions under certain conditions
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: 6, 7
Qualitative Results: NA
Comment IDs: 19, 97, 127, 139
Explanation: Under certain conditions, the interface loses data, program, or task definition. The known circumstances include:

- saving the definition under the name "New…"
- selecting a different definition from the listbox without saving first
- closing window without saving first

Two user also experienced an unexplained lost of data.
Recommendations: The first two known circumstances are simply bugs caused by situations overlooked during the design and implementation of the interface. The final one is suspicious because the interface was specifically tested for this. So, either the bug is subtle or the user may not have named the definition and thus the bug falls into the first category listed above.

F.2.2 Local

Finding #16: Users experienced difficulty while prioritizing tasks.
Evidence:
Heuristic Violation: Simple & Natural
Tutorial Results: 9
Exercise Results: 8
Qualitative Results: NA
Comment IDs: 26, 42, 63, 64, 102
Explanation: Users experienced difficulty while prioritizing tasks. The problem could be related to finding the task in the list of defined tasks. It could be related to the bizarre form of drag-n-drop used because as a result of the limitations of the widget library. It could be related to difficulty remembering how to select a

task to be hoarded. It could be related to difficulty unhoarding currently hoarded tasks. It could be because of a bug in the drop algorithm that decides where the task has been dropped.
Recommendations: First, we need to fix the bug in the drop algorithm that determines where the user dropped the task. It is likely that this bug is the cause of the problem. Should reports continue after the bug is fixed, we will need to reexamine the problem, perhaps by observing a few new users of the system as they learn to prioritize tasks.

Finding #17: Users have difficulty entering data into "Data Definition" window
Evidence:

> **Heuristic Violation:** Feedback
> Error Messages
> Prevent Errors (modal behavior)
> **Tutorial Results:** 6
> **Exercise Results:** 6
> **Qualitative Results:** 2, 7, 9
> **Comment IDs:** 3, 20, 36, 56, 101, 132, 196

Explanation: Users experienced two problems while defining user data elements. The first of these problems occurred when users entered a non-existent directory into the pathname entry area of the window. When this happened, users expected the meta-information area to become enabled (because they had entered the name of a directory). Because the system could not verify that the pathname entered was a directory, it did not enable the directory. There was no other feedback as to the problem. Indeed, the interface didn't even recognize the problem because it assumed the pathname to be a file. The second problem occurred when users tried to specify meta-information using <Tab> and <Enter>. Users were confused about the effect of these keys and were unsure about when each was appropriate to use.
Recommendations: The first problem is easy to remedy. The interface must check that the input pathname exists before checking to see if the input pathname is a directory. If the pathname does not exist, the interface must indicate this information to the user. The indication should take the form of a pop-up dialog box initially, but the interface might allow users to customize the indication to a beep or something similar. The second problem is more perplexing. The cause of the confusion is unclear to me. The <Tab> key advances the highlighting to the next area (as it does in many other windows and in other systems). The <Enter> key selects the current area. The solution may be to abandon these accelerators as "problem causing" and find different ones that cause less confusion. Users made a number of suggestions (indirectly by saying that they "expected" a certain key to have some effect). These suggestions included:

- Using <Back-Tab> to go backwards[6]
- Using the arrow keys to move around the meta-information[7]

One user also expected ^U to delete an entire pathname. Users also wanted to be able to click in the disabled meta-information area, though this desire may have been because they expected the meta-information to have been enabled for non-existent directories.

◆ **Finding #18:** Interface loses pathnames
Evidence:

> **Heuristic Violation:** NA
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 107

Explanation: The interface occasionally loses a pathname after the user types it and hits <Return> in the "Program Information" window.
Recommendations: Fix the bug.

---

[6] Note that this expectation suggests that at least one user found the <Tab> intuitive.
[7] The right arrow key, then, would work like the <Tab> key does in the current implementation; the left arrow, then, like <Back-Tab> would.

Finding #19: Users experience difficulty finding an element in predefined list
Evidence:
Heuristic Violation: NA
Tutorial Results: 4
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 18, 110, 123
Explanation: Users have difficulty finding an element in the predefined list.
Recommendations: It remains unclear how to layout this window to make it easier to find definitions. One option would be to make the window larger, allowing more of the predefined list to be visible at one time. However, unless having a larger partial listing of all the predefine elements somehow makes it easier to orient to the entire list, this “solution” only delays the fundamental problem. The only other solution is to change the way in which tasks are defined, perhaps using a tree view similar to that offered in the Hoard Walk Advice Request window. Even this solution does not imply that all elements will be visible on the screen however. Addressing this problem completely is non-trivial, perhaps impossible.

Finding #20: User has difficulty unselecting element from the contains list
Evidence:
Heuristic Violation: Simple and Natural
Tutorial Results: 8
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 62
Explanation: Two users experienced difficulty when trying to remove an element from a task definition. One user guessed correctly the first time. The other guessed correctly on the second attempt, but deleted an entire task with his first guess.
Recommendations: Redesigning the “Task Definition” to use a tree view with check boxes would probably solve this problem (see Finding #19 above). Alternatively, an undo facility would help those users who guess incorrectly (see Finding #12 above).

◆ **Finding #21:** Submitting a password takes excessively long.
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 104
Explanation: After the user enters a password to the Authentication Information window, the completion of the authentication protocol takes excessively long (on the order of a minute).
Recommendations: This problem is not local to the interface. The interface uses the “clog” program to perform authentication. It is this program that experiences the delay. Coda developers are aware of the problem.

F.3 Level 3

A *level 3* usability problem has only a minor effect on usability.

F.3.1 Global

◆ **Finding #22:** User believes colors mean too many different things
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 262
Explanation: User believes that colors in the interface mean too many different things. In particular, the warning urgency color (yellow) is used to mean both "busy" and "warning".
Recommendations: The interface should introduce a distinct color to mean "busy", perhaps a brighter shade of green.

◆ **Finding #23:** Interface offers no feedback to indicate a window is already visible
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 208
Explanation: When the user double clicks on an indicator whose window is already visible, the interface does nothing.
Recommendations: The interface should raise the window in question.

Finding #24: Windows do not resize gracefully.
Evidence:
Heuristic Violation: Consistency (internal)
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 4
Explanation: When users enlarge certain windows, those windows do not take advantage of the extra screen real estate.
Recommendations: This is a bug in the interface that should be trivial to correct.

Finding #25: Users confused when help windows grab and maintain focus.
Evidence:
Heuristic Violation: Memory Overload
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 13
Explanation: When the user clicks on a HELP button, the system displays a new window that grabs and maintains the input focus. Users who attempt to input data in a definition window or change the configuration of an event while a help window is visible find that they cannot do so until they close the button. No indication is given to them that this is the cause of their problem, though most (if not all) of them figured it out with little problem.
Recommendations: The help windows should not grab or maintain the input focus.

Finding #26: Users confused that changing name and saving creates a new definition.
Evidence:
Heuristic Violation: NA
Tutorial Results: 8
Exercise Results: NA
Qualitative Results: 15
Comment IDs: 5
Explanation: When working with any of the three definition windows (Data, Program, or Task), changing the name of the current definition and saving creates a new definition with the specified name.
Recommendations: Correcting the "Save" behavior as described in Finding #13 above will also solve this problem.

Finding #27: Users dislike pathnames entry.
Evidence:
Heuristic Violation: Memory Overload
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 11
Comment IDs: 21, 94
Explanation: Users comment that they dislike entering pathnames and would prefer to browse the file system hierarchy. Users also comment that they are concerned that the entire pathname is not visible on the screen.
Recommendations: The interface could allow users the option of typing in the entire pathname or browsing the file system to find the object for which they are looking. The concern over having the entire pathname not visible is more problematic since pathnames can be very long and screens are just a finite width. Once browsing is added, the final component could be visible in its entirety, but that does not solve the problem. This problem applies to both the "Data Definition" and "Program Definition" windows.

◆ **Finding #28:** Hoard Walk and Advice indicators confused.
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 10
Explanation: Two users were confused by the distinction between the Hoard Walk and Advice indicators. This confusion results from the Hoard Walk Advice Request being accessible from either indicator.
Recommendations: The Hoard Walk Advice Request should be accessible from only the Advice indicator. The Hoard Walk indicator should change to a red meter to indicate that the hoard walk has stalled.

Finding #29: Hoard Walk and Task indicators confused.
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 4
Comment IDs: 31, 126
Explanation: Two users were confused by the distinction between the Hoard Walk and Task indicators. My hypothesis is that this confusion occurs because the Task indicator allows users to select tasks for hoarding and because all other hoard-related information appears under the Hoard Walk indicator.
Recommendations: Moving the task selection process into the Hoard Walk indicator seems like a reasonable idea at first glance. However, such an approach would cause unexpected consequences in the use of the Task indicator light to notify users of "Task Available" and "Task Unavailable" events. When

one of these two events occurs, the Task indicator light changes colors (or otherwise notifies the user of the event). If the user double-clicks on the indicator light, hoarded tasks that are available or unavailable are shown in the appropriate color. Without the hoarded task list in the "Task Information" window, this second-level of information would be meaningless. Moving this second-level information into the list of tasks would make it less obvious to users that only hoarded tasks are marked "available" or "unavailable", introducing a different and potentially more problematic type of confusion. Because of the potential for introducing more confusion, extreme care must be taken when addressing this problem.

Finding #30: User experiences window overload
Evidence:
> **Heuristic Violation:** Memory Overload
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 24

Explanation: Four users commented that they were experiencing window overload. At the time, users had multiple windows open simultaneously.
Recommendations: Although the interface certainly allows users to get into a state of window overload, it does not require them to do so. Users control how many windows are open at any given time. The interface "requires" them to have three windows open simultaneously during the process of creating a task definition. For instance, a user might have the "Task Information" window opened, a "Task Definition" window opened, and a subordinate "Data Definition" window opened. Users can "shoot themselves in the foot" by opening more definition or other informational windows, but they may also find the ability to do so useful.

Finding #31: Interface mixes upper- and lower-case letters.
Evidence:
> **Heuristic Violation:** NA
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 7

Explanation: The interface mixes upper- and lower-case letters in certain lists.
Recommendations: This is a bug. The interface should not do this. The only exception is the "New…" element, which should always appear first.

Finding #32: User confused by terminology
Evidence:
> **Heuristic Violation:** Speak User's Language
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 209-234

Explanation: Various users report confusion over the meaning of a number of different terms. Of these reports, just one was reported by more than 2 users. Four users expressed confusion over the term "subtasks" to mean "tasks". It is interesting to note that the terms were chosen based upon an analogy to directories and subdirectories. Three users expressed confusion by the distinction between "hoarded" and "available".
Recommendations: Although most of the terminology comments were made by just a single user, it seems prudent to have a technical writer examine the interface, paying particular attention to those 25 terms, and suggest improvements.

Finding #33: Meters shown at 1% have no visible color.
Evidence:

Heuristic Violation: Feedback
Prevent Errors
Consistency (internal)
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 8

Explanation: Meters shown at low, positive percentages have no visible color, though meters shown at 0% have a small sliver of color.
Recommendations: The 0% meters have been handled specially to show a small sliver of color. Other small-valued percentages should be handled similarly.

Finding #34: <Enter> does not make the help window disappear.
Evidence:

Heuristic Violation: Consistency (with Tutorial and Exercises)
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 12

Explanation: Because of the behavior of the Tutorial and Exercises, users expect <Enter> to make the help windows disappear. Because each window of the Tutorial and Exercises had a button labeled OK and because <Enter> acted as a keyboard accelerator for the OK button on these windows, users reasonably expect <Enter> to be the keyboard accelerator for help windows, which also have a button labeled OK on them.
Recommendations: Add the keyboard accelerator.

Finding #35: Numbers are sometimes unreadable on meters
Evidence:

Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 77

Explanation: When meters are almost full, the numbers overlap with the edges of the meter, making them difficult to read.
Recommendations: This is a bug.

Finding #36: Users confused by addition of empty entry widget
Evidence:

Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 37

Explanation: When the user enters an element into either a program or a data definition, the interface adds an empty entry widget to the end of the definition. The purpose of the empty entry widget is to allow the user to enter the next element. The addition of this widget confuses users, sometimes causing them to do extraneous saves.
Recommendations: If it were possible to redesign the program and data definition windows to work without having to automatically add the empty entry widget and without requiring the user to request

another widget, the problem would probably disappear. I do not immediately see a clean way of doing this under these constraints.

F.3.2 Local

Finding #37: User surprised indicator lights are dynamic
Evidence:
- **Heuristic Violation:** Consistency (external)
- **Tutorial Results:** NA
- **Exercise Results:** NA
- **Qualitative Results:** NA
- **Comment IDs:** 159

Explanation: Because the interface draws on a static metaphor and because the tutorial didn't draw the line of departure from that metaphor, one of the pilot users was surprised that the indicator lights were dynamic.
Recommendations: No real recommendation except that this topic should be addressed explicitly in any indicator help window and possibly in balloon help. After the Tutorial was revised, no other users commented about this issue.

Finding #38: User confused by the behavior of the Hoard Walk indicator
Evidence:
- **Heuristic Violation:** Consistency
- **Tutorial Results:** NA
- **Exercise Results:** NA
- **Qualitative Results:** NA
- **Comment IDs:** 10, 11

Explanation: The Hoard Walk indicator will pop-up different windows depending upon the current system state. If the hoard daemon is not active, the Hoard Walk indicator will pop-up a window displaying the time that the next hoard walk is scheduled to begin. If the hoard daemon is currently running, the indicator will pop-up a window showing the progress of the current hoard walk. If the hoard daemon is currently waiting for advice, the indicator will pop-up a window presenting the advice request. Also, both the Hoard Walk and Advice indicators can pop-up the Hoard Walk Advice Request window when such a request is pending.
Recommendations: The windows associated with the Hoard Walk indicator exhibit modal behavior. The primary problem is that the Hoard Walk Advice Request window is vastly different from the other windows associated with this indicator light. If this request were to come in only under the Advice indicator (as suggested in **Finding #10** above) and if the progress meter were to turn red until the advice request were answered, user confusion could be lessened.

Finding #39: Users confused by two-color blinking.
Evidence:
- **Heuristic Violation:** Memory Overload
- **Tutorial Results:** NA
- **Exercise Results:** NA
- **Qualitative Results:** NA
- **Comment IDs:** 9

Explanation: When an event arrives that is configured to flash the indicator light, the interface blinks between the old urgency color and the new urgency color for 10-15 seconds. Users find this flashing confusing because, unless they happen to remember what the old urgency color is, the new urgency color isn't obvious until the blinking stops.
Recommendations: One suggestion is to blink between dark gray and the new urgency color. This makes the new color obvious at the expense of forgetting the previous color. A second suggestion is to show the

new urgency color for longer periods of time and the old urgency color for shorter periods of time. Unfortunately, this second suggestion still doesn't make it entirely clear that the color shown longer is the new one. The former suggestion has been implemented.

Finding #40: Users somewhat confused because the indicator stops blinking.
Evidence:
> **Heuristic Violation:** NA
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 33

Explanation: Users become somewhat confused when the indicator stops blinking. One user wondered whether or not this behavior meant that the problem had been solved.
Recommendations: Overall, this problem presents only a minor concern. Even though six users commented that the flashing had stopped, only one of them seemed confused by the behavior. Two of them continued their comment saying that they guessed that the flashing was simply intended to draw their attention. The user who seemed confused happened to double-click on the indicator light just as it stopped flashing. He then continued the commented saying that "the impression [he is] left with is that, by opening this window, somehow the danger has just gone away", at which point he chuckled. Although the timing here was unfortunate, I don't think he was terribly confused. Perhaps the best way to address this potential source of confusion is with an explanation in the help window of the Control Panel. Another "solution", which would likely serve to introduce a more serious usability problem, would be to have the indicator blink until the user investigated the problem, forcing users to investigate a problem that they may not wish to investigate and possibly encouraging the misconception that double-clicking fixes the problem.

Finding #41: User disagrees with default notification for certain events
Evidence:
> **Heuristic Violation:** NA
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 257, 260

Explanation: Three users commented that they thought the "Token Expiry" event should be notified at a critical urgency level.
Recommendations: Because the user can modify the way in which event notification occurs, the defaults need not be changed. However, we may want to consider adding an event that warns users prior to token expiration (perhaps an hour or so beforehand). Such an event could be notified at the warning urgency level. We could then change the "Token Expiry" event to notify at the critical urgency level with no pop-up, flash, or beep and the "Activity Pending Tokens" event to notify at the critical urgency level with beeping.

Finding #42: Users search for a way to configure events
Evidence:
> **Heuristic Violation:** NA
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 119, 147

Explanation: Five users had difficulty remembering how to configure events. One of those users did not recognize the Event Configuration tab as a way to do so.
Recommendations: I suspect that this problem is a learning problem. During the Tutorial, users were lead through the process of configuring an event step-by-step. In the Exercises, they were asked how an event was configured. I'm not sure how to make finding the location of event configurations easier.

Finding #43: Users unsure how an event will be notified.
Evidence:
Heuristic Violation: Feedback
Speak User's Language
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 38, 118, 164, 170, 206, 221
Explanation: While configuring events, users were frequently confused as to how an event would be notified. For example, five users were confused as to how the system would notify an event if all the notification options were turned off. They did not realize that the minimal notification was to simply change the color of the indicator light. One user was confused as to how the indicator light would change colors. One user was uncertain what exactly was going to "flash" (another was confused as to what "flash" meant). One user suggested that the configuration tab should contain a copy of the indicator lights and another user suggested that it show an example notification.
Recommendations: The two suggestions listed above are both reasonable. If the configuration tab contained a copy of the indicator lights, it could use them to indicate which indicator would notify the user of the event (perhaps with an arrow). And, it could have a button the user could press that would show how the event is currently configured to notify users. An alternative design would be for the system to continuously notify this event on the indicator lights shown on this tab, but this would be highly annoying if the user had the system configured to beep.

Finding #44: User assumes system uses a 24-hour clock.
Evidence:
Heuristic Violation: Speak User's Language
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 195
Explanation: In the Authentication Information window, the system includes the time at which the user's tokens are expected to expire. This time is presented in military time. Two users commented that they were going to assume military time, but expressed some uncertainty about the correctness of this assumption.
Recommendations: The interface could allow users to specify whether they would prefer a.m./p.m. designations or military time. This specification belongs within the Control Panel on a new tab called "Interface Configuration" (or something similar). Within that tab, users should find another notebook that includes a "Time" tab. A simple radiobutton would work well. Alternatively, the system could always present a.m./p.m. designations because such a format leaves no room for uncertainty.

Finding #45: Users confused by the meters in the "Space Information" window
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: 2, 10
Qualitative Results: NA
Comment IDs: 270, 271, 280
Explanation: A total of four users had difficulty with the meters on the "Space Information" window in one way or another. Three of these users answered the relevant exercise incorrectly. One of these users and the fourth user commented that they were confused by the meanings of the meters. After consulting the help window, the fourth user answered the relevant exercise correctly. I'm not certain if the other users consulted the help window or not.
Recommendations: Although it is clear that the meters in this window generally confused users, it isn't clear to me how to correct the problem.

Finding #46: User confused by "questionnaire already running" dialog
Evidence:
Heuristic Violation: Error Messages
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 200
Explanation: When the user attempts to open an advice request which they have already opened, the interface pops up an error message dialog that explains that the advice request is already running. This user did not understand that dialog.
Recommendations: The explanation in the dialog may need clarification.

Finding #47: The Tix tree widget requires three clicks to expanding/contracting nodes
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: 4
Qualitative Results: NA
Comment IDs: 179, 204
Explanation: To expand or contract a node in the Tix tree widget, the user must click three times on the small checkbox widget. Furthermore, the checkbox widget has an extremely small active area.
Recommendations: The triple-click behavior was a bug in the Tix tree widget that disappeared once we upgraded to a more recent version of the Tix widget library. Furthermore, the checkbox behaves more rationally and in such a way that the small active area is not as annoying.

Finding #48: User dislikes having fetch estimates expressed in seconds
Evidence:
Heuristic Violation: Speak User's Language
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 13
Comment IDs: 96
Explanation: The hoard walk advice request lists the estimated fetch time of each object in seconds. When expected fetch time is high, the number quickly becomes meaningless to users.
Recommendations: The estimated fetch time should be listed in minutes and seconds.

Finding #49: User confused by "unselect" dialog
Evidence:
Heuristic Violation: Error Messages
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 140
Explanation: The Hoard Walk Advice Request window presents this dialog to the user if the user unselects a definition (for fetching) that is included by multiple tasks. In this dialog, the interface gives the user the choice of unselecting a task to be fetched from a single task or from all tasks. The problem arises because some elements may be listed under different tasks. The dialog confused two users who stumbled across it.
Recommendations: This situation is difficult to handle. The initial recommendation would be to clarify the explanation in the dialog and then enter a period of "watchful waiting". If further symptoms of the problem appear, the problem will need to be reconsidered.

Finding #50: Task definitions unnatural
Evidence:
Heuristic Violation: Simple & Natural
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 157, 183
Explanation: Users find it unnatural that tasks are the only definitions that can be hoarded directly. This requirement means that users must create a task definition to hoard a single file or directory. Furthermore, to examine a data or program definition, they must first examine a task definition.
Recommendations: Any redesign of task definitions should consider this issue, but there is no obvious solution within the current design.

Finding #51: Users confused by the cache meter in the "Task Information" window
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 269
Explanation: One user wondered whether this meter meant that 97 MB (the size of the Venus cache) was available or whether 70% (the percentage shown on the meter) of that 97 MB was available. Another user also thought that the meter was showing the percentage of the cache currently available. The third user was completely confused by all of the meters in the "Task Information" window.
Recommendations: In fact, not one of these users was correct. The cache meter that appears at the top of this window shows what percentage of the cache is dedicated to hoarded objects. The help dialog for this window needs to be examined to make sure that the text is clear on this point. Another recommendation would be to add a label to this meter to inform users what the meter is showing.

◆ **Finding #52:** Newly hoarded task may not be visible in list of hoarded tasks
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 43
Explanation: After a user hoards a task, that task may not be visible in the list of hoarded tasks in the "Task Information" window.
Recommendations: The interface needs to make newly hoarded tasks visible.

Finding #53: Users confused by the task meters.
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 268
Explanation: Users express confusion over the meaning of the task meters in the "Task Information" window. This comment showed up most frequently during the Tutorial phase of the study, but occasionally came up during the Exercises. Users were able to use the help window to understand the meaning of the meters.
Recommendations: Labeling the columns of the hoarded task list is likely to alleviate this confusion. For this column, the label "Percent Available" seems appropriate.

Finding #54: User finds it strange that a single click selects an item for the contains list.
Evidence:
Heuristic Violation: Consistency (internal and external)
Tutorial Results: 8
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 23
Explanation: Almost half of the participants commented that they thought it strange that single clicking on an element in the predefined list of the "Task Definition" window selected it into the contains list.
Recommendations: Because double-clicking on an element opens it, we are not left with much of an alternative. If we were to change the interface to provide a tree-based task definition procedure (see Finding #19 above), this concern would be moot.

Finding #55: Interface double-selected elements under certain circumstances
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 22
Explanation: Under certain circumstances, the interface will double-select an element from the predefined list into the contains list of the "Task Definition" window.
Recommendations: This behavior was clearly a widget bug that disappeared upon upgrading to a more recent version of tcl/tk/tix.

Finding #56: Extraneous underlining appears in lists of the "Task Definition" window
Evidence:
Heuristic Violation: Simple & Natural
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 181
Explanation: The "Task Definition" window underlines entries in the contains and predefined lists in an arbitrary way. I suspect the underlying widgets consider these underlines to be a feature, but they confused users.
Recommendations: We could remove the underlining from our versions of the widget, but that would introduce portability problems.

Finding #57: Priorities need to be labeled
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 82
Explanation: In the hoarded task list of the "Task Information" window, the columns need to be labeled. In particular, the priority column needs to be identified as such.
Recommendations: Add a line containing labels for each column.

Finding #58: Meter may hide long task names
Evidence:
Heuristic Violation: Speak the user's language

Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 85

Explanation: The meter used to show the availability of hoarded tasks may hide the end of long task names.
Recommendations: The task name needs to be a read-only entry widget that the user can scroll around in.

Finding #59: User types a pathname into the name of a program definition
Evidence:
Heuristic Violation: Speak the user's language
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 137, 198

Explanation: Three users typed pathnames into the name entry widget of a program definition.
Recommendations: Users seemed somewhat confused by program definitions. Perhaps because the name does not imply that a "program" can contain more than one program so they thought that the name of the program was the pathname to the executable?

Finding #60: User doesn't believe system verifies existence of executable programs
Evidence:
Heuristic Violation: Prevent Errors
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 61

Explanation: User does not believe that the system verifies the existence of programs.
Recommendations: User is correct. The system does no such verification. In the interests of preventing misunderstandings, the system should verify their existance.

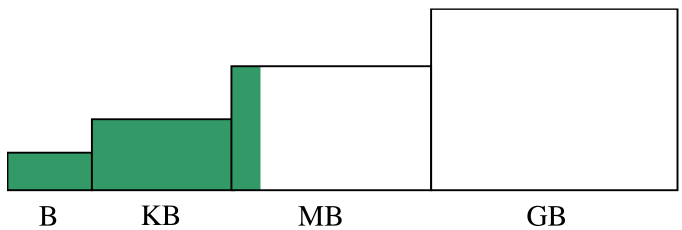
F.4 Level 4

A *level 4* usability problem is more subtle and often suggests a future enhancement.

F.4.1 Global

Finding #61: Meters do not cover orders of magnitude or relative information very well
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 50, 51

Explanation: Two sets of meters in the interface are required to cover multiple orders of magnitude. One set appears in the "Space Information" window and must cover a few megabytes of space to a few gigabytes of space. Another set appears in the "Network Information" window where the system must show differences between kilobytes (or no) network bandwidth and hundreds (or more) of megabits of bandwidth. As currently designed, the meters shown in these windows do a poor job of presenting differences that span orders of magnitude. Furthermore, they are unable to indicate how close the system is to exceeding a threshold or how far beyond the threshold the system has wandered.
Recommendations: One possible redesign the meters would be to use a pseudo-log scale applied to meters. Another possibility is to use meaningful breakpoints rather than log-scale breakpoints. For example, to use just one cut-off that corresponds to the threshold between strong and weak connectivity. Such a scheme would give users a visual indication regarding how close current network conditions are to the threshold.



Finding #62: User requests balloon help
Evidence:

> **Heuristic Violation:** Help
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 90

Explanation: User requested balloon help be added to the interface.
Recommendations: Excellent area for improvement.

Finding #63: User believes task interface to be overly complicated
Evidence:

> **Heuristic Violation:** Memory Overload
> **Tutorial Results:** NA
> **Exercise Results:** NA
> **Qualitative Results:** NA
> **Comment IDs:** 203

Explanation: User believes the task interface to be overly complicated. We have no further details as to what user meant.
Recommendations: Any redesign of the task definitions needs to consider simplifying the definition process. A tree view of tasks (see Finding #19 above) might help.

F.4.2 Local

◆ **Finding #64:** Interface is cognizant of vertical screen real estate, but not horizontal
Evidence:

> **Heuristic Violation:** NA

Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 169

Explanation: The interface takes pains to reduce the amount of screen real estate required by the indicator lights. The indicator lights require just 1"x2" of the screen. However the interface does not provide a means for displaying the indicator lights horizontally across the top of the screen. Such a layout could require 9"x¼" of the screen. Users may be more willing to give up ¼" across the entire width of the screen than a block along one side.

Recommendations: The interface should allow a horizontal layout of the indicator lights.

Finding #65: Users would like help on the indicator lights window.

Evidence:

Heuristic Violation: Help
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 45

Explanation: Three users comment that they were surprised no help was available for the indicator lights themselves.

Recommendations: Adding a Help indicator might be appropriate.

Finding #66: User concerned about missing an event notification

Evidence:

Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 16
Comment IDs: 41

Explanation: Two users were afraid of missing an event notification.

Recommendations: One suggested that the interface make it clear what is new since the last time the user looked at the indicators. Although this would be nice, it is a tad difficult for the interface to determine when the user last looked at the indicators—short of installing specialized hardware to monitor the user's eye movements. At best, we could make it clear, perhaps with an asterisk, what is new since the last time the user opened an indicator window.

Finding #67: User would like feedback regarding the "health" of Venus

Evidence:

Heuristic Violation: Feedback
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 25

Explanation: The user would like the interface to provide feedback regarding the "health" of Venus. By this, the user could mean many things. One possibility is whether or not Venus is "alive".

Recommendations: The interface could indicate to users that its connection to Venus is alive by using all of the indicator lights. If the interface detects that its connection to Venus has disappeared, it could indicate this by turning all indicator lights dark gray. Once the connection is reestablished, the indicator lights could go back to their actual urgency state.

◆ **Finding #68:** User would like different configuration settings.

Evidence:

Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 187

Explanation: User suggested that the system allow different configuration settings. For example, he'd like to have one set of configurations for use "at school", a second one for use "at home", and a third set for use "on the road". His reasoning is that when he is at school, the fact that he is disconnected is interesting and important news. When he's at home, it's the norm.

Recommendations: This suggestion seems to be a reasonable request. The Control Panel will need an additional tab that allows the user to switch between different configurations easily.

Finding #69: User would like a set of system defaults

Evidence:

Heuristic Violation: Clearly Marked Exits
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 145

Explanation: User suggests having a set of system defaults that the user could revert back to if necessary.

Recommendations: Such a set of system defaults would give new users confidence to explore different configurations and seems reasonable.

◆ **Finding #70:** User wants to modify all event notifications at once

Evidence:

Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 87

Explanation: Two users wished for the ability to modify all event notifications at once. Such a facility would be particularly useful for users who would like to turn beeps or pop-ups off all at once.

Recommendations: Extending the interface in this way would provide a modest improvement in usability, but other changes should take precedence.

Finding #71: User would like to be able to stop long transfers over weak networks

Evidence:

Heuristic Violation: Clearly Marked Exits
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 18
Comment IDs: 178

Explanation: When the system fetches an object over a weak network connection, users would like to have the ability to abort the fetch.

Recommendations: This suggestion, though a very reasonable request, is difficult to implement with the current RPC2 package. To abort a fetch, Venus would have to completely tear down the RPC2 connection and re-establish it. In the long-term however, users need this feature.

Finding #72: User would like a count-down timer on advice requests

Evidence:

Heuristic Violation: Feedback
Tutorial Results: NA

Exercise Results: NA
Qualitative Results: NA
Comment IDs: 68

Explanation: User would like to see a count-down timer on any advice requests that timeout after a given amount of time.
Recommendations: This is a reasonable request.

Finding #73: User wants expert help for Space problems
Evidence:

Heuristic Violation: Help
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 76

Explanation: User wants the system to explain how to solve space problems. In particular, the user wants the system to explain how to reinitialize Venus, how to increase RVM size, and how to increase cache size.
Recommendations: If the user were to reinitialize Venus while operating disconnected, they would lose any modifications made after the point of disconnection. Because of the potential for data loss, the help window simply instructs users to contact their system administrator. Although it would not be difficult to provide instructions for any of these procedures, I question the wisdom of doing so.

Finding #74: The Network help screen should explain why the network is degraded
Evidence:

Heuristic Violation: Help
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 250

Explanation: User wants the Network indicator's help window to explain why the network is down or weak.
Recommendations: Although this suggestion sounds spectacular, the request is unreasonable because it is unlikely that the system will know why the network connection is down or weak. To the extent that the system knows this information, it should make the information available.

Finding #75: User wants the Task help screen to explain why a task is unavailable
Evidence:

Heuristic Violation: Help
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 252

Explanation: User wants the Task indicator's help window to explain why a task is unavailable.
Recommendations: Again, to the extent that the system knows why a task is unavailable, it should make that information available to the user.

Finding #76: System should predict if hoarding a task would cause space problems
Evidence:

Heuristic Violation: Minimize User's Memory Load
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA

Comment IDs: 15, 273

Explanation: When the user selects a task to be hoarded, the system should predict whether or not doing so will cause a space problem. If it would, the system should warn the user. Furthermore, the system should calculate how much additional space would be required to hoard everything the user has requested.

Recommendations: These suggestions are both good ideas for future extensions. However, there are limitations to the system's ability to predict space problems. The space estimates for tasks are only estimates, based upon the last hoard walk. The next hoard walk may discover that the size of a task has either increased or decreased. Within the limitations of the system's knowledge, the interface should certainly be able to offer this information to the user.

Finding #77: User would like a tree view of tasks

Evidence:

Heuristic Violation: Consistency
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 110

Explanation: User suggests using a tree view in the "Task Definition" window similar to that used in the Hoard Walk Advice Request window.

Recommendations: Although this suggestion is appealing, its implementation involves some rethinking of the task definition. One stumbling block is how users would specify meta-information for data directories (e.g. immediate children vs. all descendants). Overall, this suggestion sounds like a good idea, but will require careful thought.

Finding #78: The "Data Definition" window should indicate the size of subtrees.

Evidence:

Heuristic Violation: Memory Load
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 57

Explanation: In the "Data Definition" window, the size of each subtree should be indicated to help users determine the appropriate meta-information.

Recommendations: Adding this information appears at first glance to be a very simple way to improve the usability of this interface. Unfortunately, obtaining the information could be somewhat tricky. The information is not maintained dynamically so it would have to be calculated, requiring a recursive walk down the tree. Such a walk would require us to fetch all of the data contained in the tree and could have far-ranging effects by triggering resolutions. Getting this information to the interface is an area to explore as it would certainly help users make these decisions, but it will require care to implement efficiently.

Finding #79: User wants more from program definitions

Evidence:

Heuristic Violation: Simple and Natural
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 153, 173, 190

Explanation: User wants to be able to see the details of program definitions and expects program configuration files (*e.g.* `.cshrc` or `.twmrc`) to appear in program definitions automatically.

Recommendations: Both requests seem to be reasonable extensions to the current system, though showing the details of the program definition should not be the default.

Finding #80: User expects interface to track and automatically hoard missing objects
Evidence:
Heuristic Violation: Minimize User's Memory Load
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 23
Comment IDs: 103
Explanation: Because the system asks the user about the severity of disconnected cache misses, the user expects the system to use this information to track these misses and automatically hoard the more important ones.
Recommendations: This request seems very reasonable.

◆ **Finding #81:** Bogus fetch statistics confuse user
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 99
Explanation: One user was confused by the bogus fetch statistics in the Hoard Walk Advice Request window. The Tutorial specifically told the user to ignore this area of the screen, but he evidently missed the instruction or didn't understand it.
Recommendations: Implement this feature of the window.

Finding #82: Users confused when a file becomes unavailable
Evidence:
Heuristic Violation: Feedback
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: NA
Comment IDs: 29
Explanation: At one point during the Exercises, all tasks were available. Shortly thereafter, a task became unavailable. Three users were uncertain as to why it was no longer available. Later, a different user became confused because the interface asked whether or not a file within a previously available task should be fetched over a weak connection.
Recommendations: Although the second of these two situations points to what is likely a bug in the Exercises, the first situation points to a usability problem. Users are clearly having difficulty understanding under what conditions an object (and therefore a task) can become unavailable. The problem is that the interface does not explain why something has become unavailable. The user has no idea whether it was updated at the servers or whether we simply do not know whether or not the object is valid (e.g. the client has lost its callback). The interface offers no hint as to the problem. One possible solution to the problem is to track this information by paying attention to whether we've lost contact with a server or a volume validation failed or a file validation failed. This information could then be presented to the user, perhaps if the user right-clicks on the task entry in the hoarded list. This problem and its proposed solution needs careful attention. I question the wisdom of exposing the details of cache consistency to novice users.

◆ **Finding #83:** User uncertain whether or not a program is available
Evidence:
Heuristic Violation: NA
Tutorial Results: NA
Exercise Results: NA
Qualitative Results: 10
Comment IDs: NA

Explanation: User was not certain whether or not a program was available in the cache. His uncertainty occurred in the process of providing hoard walk advice. The advice request did not mention the program in question.
Recommendations: At least part of the user's confusion can be attributed to the artificial nature of the Exercises. He had not created the task definitions that he was using in the advice request so he didn't have the information needed to realize whether or not the task definition contained the program. Because the hoard walk advice request only lists those objects not in the cache, he was unable to use his background knowledge to deduce that the program was available. The question that remains is whether or not this is a good model. Even users who created the original definitions are likely to forget them over time. The design should be reevaluated with this information in mind to determine if there is a clean way to add this information to the request without overloading the user's ability to comprehend the information available.

Finding #84: Users express concern over difficulty knowing what data is required
Evidence:
- **Heuristic Violation:** NA
- **Tutorial Results:** NA
- **Exercise Results:** NA
- **Qualitative Results:** NA
- **Comment IDs:** 185

Explanation: Users express concern over the difficulty in knowing precisely which files and programs are necessary to work on a given task.
Recommendations: Determining what information to hoard is, indeed, a difficult task. Any additional help the system can provide is well worth the effort. For instance, the interface could track what programs access data the user has hoarded and offer hints if the user has not also hoarded those programs.

Finding #85: User would like a graphical representation of what is in the cache
Evidence:
- **Heuristic Violation:** NA
- **Tutorial Results:** NA
- **Exercise Results:** NA
- **Qualitative Results:** 10
- **Comment IDs:** 205

Explanation: One expert user expressed uncertainty over whether a particular program was available in the cache. Another expert user suggested that the system offer a graphical representation of what is currently cached.
Recommendations: I have mixed opinions regarding the value of such a tool. On the one hand, I see it as exposing too much information regarding caching—in essence, it makes caching opaque. On the other, I see that it offers a secondary way for users to determine what is and is not available for use during disconnected operation. As a tool for developers and experts (the user making the suggestion fell into this category), it would certainly be valuable.

◆ **Finding #86:** User would like a visual representation of Venus' on-going activities
Evidence:
- **Heuristic Violation:** NA
- **Tutorial Results:** NA
- **Exercise Results:** NA
- **Qualitative Results:** NA
- **Comment IDs:** 243

Explanation: One user suggested a visual representation of Venus' on-going activities, akin to the current mariner shown in the codacon. In essence, the user has requested that the codacon be incorporated into the new interface.
Recommendations: Again, I have mixed opinions regarding the value of such an extension. The codacon is a useful tool for developers and even expert users. Unifying the two tools makes the developers job easier at the expense of adding additional information into a user interface intended to help novices. The

information provided in the codacon is not something that novice users should be exposed to. Yet, should they run into a problem, the information in such a tool would be useful to system administrators and developers. If such an extension is added to the interface, it must be clearly marked as being intended for developers and system administrators.

Finding #87: Users may feel limited by constraint that only tasks can be hoarded
Evidence:
> **Heuristic Violation:** Simple & Natural Dialogue
> **Tutorial Results:** NA
> **Exercise Results:** 14
> **Qualitative Results:** 8
> **Comment IDs:** NA

Explanation: Users had to reorganize their tasks so that they could give a priority to something they had originally included as a subtask.
Recommendations: This behavior suggests that users may feel constrained by the limit that only tasks may be prioritized. Future versions of the interface may need to reconsider this design decision.