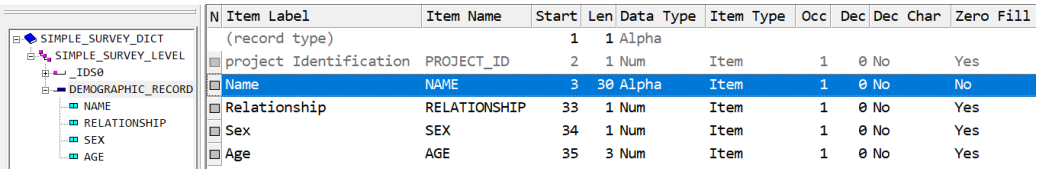
Project:
CS6750 – Human-Computer Interaction

Aaron Whitesell
awhitesell3@gatech.edu

1 INTRODUCTION

I selected an interface within Census and Survey Processing System (CSPro). The software's scope is broad, allowing users to design applications for data collection, collect data on mobile devices, synchronize to web applications, edit data, as well as analyze the data. CSPro is public domain software and has been used in over 190 countries.

The interface selected for redesign within CSPro is referred to as dictionary editor. The dictionary editor allows the user to add variables to the survey and specify valid responses. As an analogy, you can think of the dictionary editor as the functionality in PeerSurvey that allows you to add a question and select the type (short answer, choose one, choose multiple, agreement, frequency, and age group). The difference being that CSPro has additional functionality, like customizable forms and a scripting language to control behavior. Figure 1 demonstrates what a simple survey will look like when viewed in dictionary editor.



N	Item Label	Item Name	Start	Len	Data Type	Item Type	Occ	Dec	Dec Char	Zero Fill
	(record type)		1	1	Alpha					
	project Identification	PROJECT_ID	2	1	Num	Item	1	0	No	Yes
	Name	NAME	3	30	Alpha	Item	1	0	No	No
	Relationship	RELATIONSHIP	33	1	Num	Item	1	0	No	Yes
	Sex	SEX	34	1	Num	Item	1	0	No	Yes
	Age	AGE	35	3	Num	Item	1	0	No	Yes

Figure 1 — Simple survey with four variables added from the dictionary editor.

Here the variables NAME, RELATIONSHIP, SEX, and AGE have been added to the survey. The different attributes of the variables are also viewable. We can see the NAME variable has a type of alphanumeric and is of length 30. Each variable can be given a set of responses which in CSPro nomenclature is called a value set. The dictionary editor allows the user to drill down on each variable to add, edit, and delete the value set. Figure 2 shows the value set for the SEX variable.

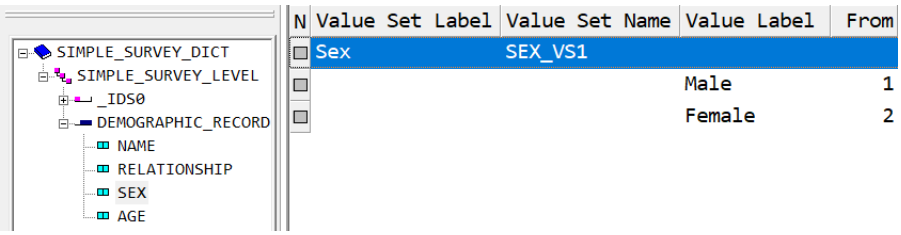


Figure 2 – View of value set from dictionary editor.

Every survey will require adding variables to the survey. In this simple survey there were only four variables. However, in more complex surveys the application developer will be working on surveys with hundreds, if not thousands of questions. A lot of the application developer's time will be spent in this interface and even small improvements will pay significant dividends.

2 INITIAL NEEDFINDING

2.1 Select approaches

I have selected two needfinding approaches. The first approach will focus on novice users of CSPro. I will review publicly available forum posts that are related to the dictionary editor. The user issues identified during this needfinding activity can then inform the second needfinding activity. The second approach will focus on expert users and developers of CSPro. Each expert will be interviewed to gather insight on the dictionary editor interface.

2.2 Plan activities

Users frequently post their CSPro issues to the community-driven website csprousers.org. Here I will perform a search of the forms for the term "dictionary editor." For each relevant search result, the user issue will be examined and considered for additional needfinding. Considerations will be frequency of issue by users and ability to positively affect issue through an interface design. Collected information will include forum title, number of posts in thread, understanding of issue, and URL.

The experts will consist of three colleagues I work with at the U.S. Census Bureau. Between the three of them, they have a combined 76 years of experience working with CSPro and predecessors of CSPro. The interviews will be conducted using Zoom. Note-taking will be used to record details of interview, because it is non-intrusive and quick to review. The interview will include

questions to glean knowledge from their experience and questions gathered from the forum review.

2.3 Report conclusions

Searching for "dictionary editor" returned 16 relevant threads. A summary of each thread can be found in *Appendix 7.1 Forum review.* Two user issues were identified from these 16 threads and included in the interview.

The first issue identified is that users are editing their variables outside of the dictionary editor. The users are doing this, because the dictionary editor can be cumbersome and slow when editing many variables. One-way users get around this is they directly edit the dictionary files which is a text file. However, this leads to an incorrectly formatted dictionary, because the CSPro editor could not enforce its validity checks. The user is now own their own to fix the dictionary which leads to many support requests.

The second issue is that users are sometimes discouraged from using subitems. The dictionary editor allows you to combine multiple variables into a single parent item while retaining the ability to access them by their subitems. This functionality can be useful, but other uses of subitems can lead to poorly designed applications. Ideally, the next needfinding activity will identify these scenarios and allow the interface redesign to constrain the use of subitems to only beneficial uses.

The interviews led to a better understanding of the strengths and weaknesses of the dictionary editor. All three interview participants agreed that the dictionary editor's interface was effective at representing the dictionary variables. Additionally, the default values for a newly added variables are effective for speeding up the task of adding variables. On the other hand, some of the weaknesses included navigating the interface and the discoverability of functionality within the interface.

The interviewees had a good understanding of why *users edit their variables outside of the dictionary editor* when working on large surveys. By editing the dictionary directly or copying pasting the contents to Excel the user is able to get able to zoom out and get a birds-eye-view of all their variables. This facilitates easy navigation from variable to variable for fast modifications. It also allows the user to take advantage of search and replace functionality.

With subitems the interviewees had very similar uses for subitems. Across the board they were used for dates, industry codes, and occupation codes. With regards to these specific uses subitems have their value. The strength of subitems can also be their weakness. They are intuitive and flexible. This leads to novice users trying to use subitems for tasks other functionality would be better suited for. Additionally, their flexibility and lack of constraints can lead to users organizing their data in a third dimension which is not currently supported by CSPro.

The interview questions and responses can be found in *Appendix 7.2 Interview questions and Appendix 7.3 Interview responses.*

3 HEURISTIC EVALUATION

Heuristic evaluation is a type of predictive evaluation where an expert would review the interface and identifies areas where the design principles are violated. I will act as the expert and identify both areas that works well and areas that do not work well.

3.1 Areas that work well and why

The dictionary in CSPro contains all the variables for a survey. The variables act as storage for each question in the survey. Similarly, to how a dictionary may consist of all the words in a language a CSPro dictionary consists of all the variables in a survey. The *design principle of mapping* explains this approach of having the system use real-world concepts to help the user understand the usage within the system. This works well, because it lowers the *gulf of execution* for the user.

The dictionary editor is effective at representing the collected data, because it in many ways resembles the collected data. Here the dictionary editor is leveraging the *design principle of structure*. This works well, because the similarity between the dictionary editor and the collected data strengthens the user's mental model which helps the user better understand how their changes in the dictionary editor affects the collected data.

When a dictionary variable is deleted the associated question, text is not deleted. This is done, because the *distance* between the dictionary variable and question text is far. Entirely different editors are used to add, delete, and modify each. Here the *design principle of tolerance* is being leveraged. This works well, because

this helps users not unknowingly delete work created in another editor and possibly by another developer.

CSPro translates their survey runtime errors into English, Spanish, Portuguese, Russian, Vietnamese, and Chinese. Here the *design principle of equity* is being leveraged. This works well, because CSPro can then accommodate a larger user base.

All shortcuts in the dictionary editor are also available in the menus. This allows CSPro to design an effective interface for both expert and novice users. Here the design principle of flexibility is being leveraged. This works well, because expert users are able to use the shortcuts and the novice users are able to search for the functionality in the menu.

CSPro and the dictionary editor have comprehensive documentation. For a technical piece of software like CSPro that encompasses so much, I believe this is necessary. Here the *design principle of documentation* is being leveraged. This works well, because the comprehensive documentation is searchable. Conveniently, if a keyword is highlighted and F1 is pressed the helps for the keyword are displayed. Also, each keyword includes an example that targets a user task.

3.2 Areas that do not work well and why

Conceptually subitems are easy to use. This leads to novice users using them in ways that can lead to a poorly designed application. I think this partly occurred, because of an *expert blind spot*. As designers and software developers who implemented the functionality, know their intended use. This does not work well, because to a novice user their purpose is ambiguous. The *design principle of constraints* was not leveraged to steer the novice user into the intended uses of subitems.

Expert users often work outside of the dictionary editor when working with large numbers of variables and value sets. The reason is a text editor or spreadsheet application will allow them to complete their work faster. The reason the CSPro editor is not as effective is because it does not give users a birds-eye-view. With the dictionary editor more mouse and keyboard actions are necessary to access each individual variable and value set. Additionally, the lack of search and replace functionality exacerbates the interface problem. I believe this has not been resolved, because we, the designers and software developers are not survey

statisticians who create surveys with hundreds or thousands of questions. This is another reminder that *you are not the user*.

The dictionary editor allows the user to select a variable which includes a row of options. Once the cursor is in a cell it can edit the field. However, the *design principle of consistency* is not being leveraged. Some fields are text cells while others are drop down cells. If the cursor is editing a text cell and the user presses the down arrow the variable below it will be selected. If the cursor is in a drop-down cell and the user presses the down arrow the contents of the drop down will be displayed. See figure 3 for an example of this scenario.

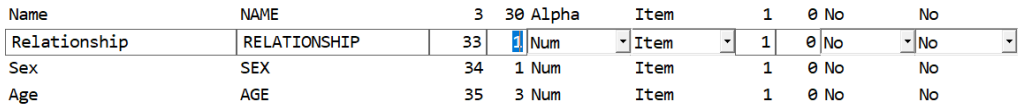


Figure 3 – Example of editing a text cell.

This behavior is not consistent between cells or between applications. Excel also displays information in tabular fashion, but allows cells to be consistently navigated by using the directional arrows. Accessing the contents of a drop-down cell is done by pressing ALT + down arrow.

The dictionary editor includes a panel on the left-hand side that include a file tree, dictionary tree, and a form tree. This allows the user to work in the dictionary editor, but be inspecting the form tree. This can be convenient, because it makes information from another editor easily accessible. However, each tree looks relatively similar. With novice users this can lead to *knowledge-based mistakes*. This is a novel interface and novice users will incorrectly assess the state of the interface when they are learning to use the interface. With expert users this can lead to *action-based slips*. Often without thinking about it an expert user can select the correct option in the incorrect tree. See figure 4 shows the tabs for the tree panel.

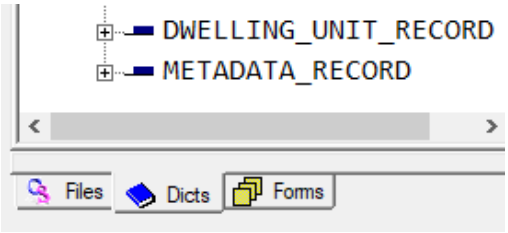


Figure 4 – View of the tree tabs that can cause both novice and expert users' confusion.

4 INTERFACE REDESIGN

4.1 Dictionary editor with grid interface

The interface redesign of the dictionary editor focused on the ability to view, add, edit, and delete variables from within it. See figure 5 for a wireframe prototype of the entire dictionary editor interface.

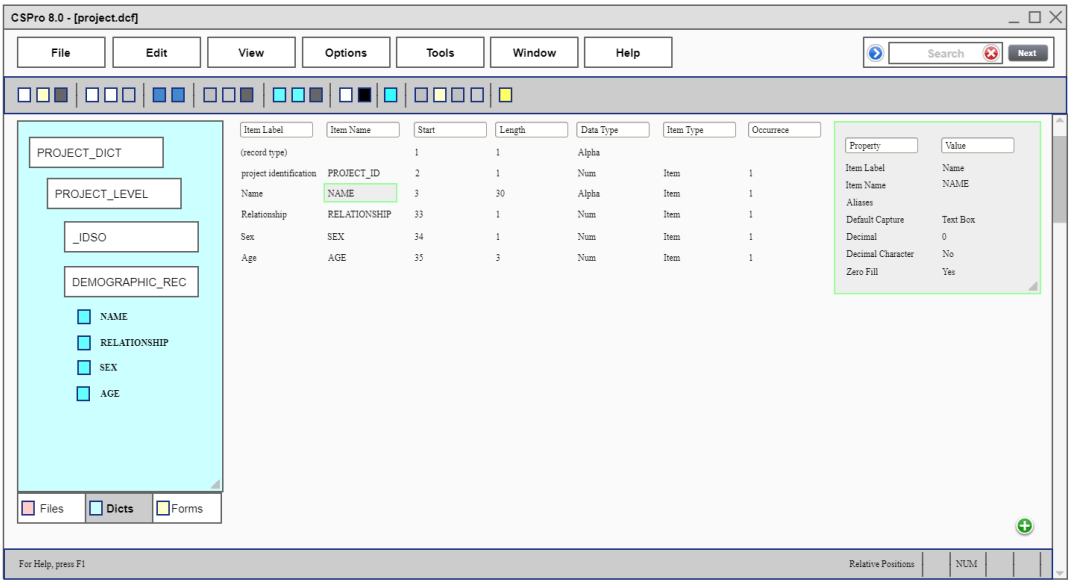


Figure 5 – View of redesigned dictionary interface.

Previously, selection of a row in the dictionary editor highlighted the entire row. Each cell had to be click again to edit. Now the selected cell is highlighted and editing is enabled without having to click into the cell. See figure 6.

Item Label	Item Name	Start	Length	Data Type	Item Type	Occurrece
(record type)		1	1	Alpha		
project identification	PROJECT_ID	2	1	Num	Item	1
Name	NAME	3	30	Alpha	Item	1
Relationship	RELATIONSHIP	33	1	Num	Item	1
Sex	SEX	34	1	Num	Item	1
Age	AGE	35	3	Num	Item	1

Figure 6 – Close-up view dictionary variables represented in rows with a single cell selected.

In the top right of the interface a search bar was added. The search bar becomes visible when a cell of a dictionary variable is selected. The red circle clears the

text string, while the blue circle expands the search interface to include search and replace functionality. See figure 7.

Figure 7 – Close-up view of the search bar.

When the search bar is expanded the search and replace functionality is exposed. The expander icon rotates 90 degrees and the replace text box appears below the original search text box. See figure 8.

Figure 8 – Close-up view of the search bar expanded to allow search and replace.

When a cell of a dictionary variable is selected a grid on the right-hand side appears. It displays "extended" attributes for each selected dictionary variable. See figure 9.

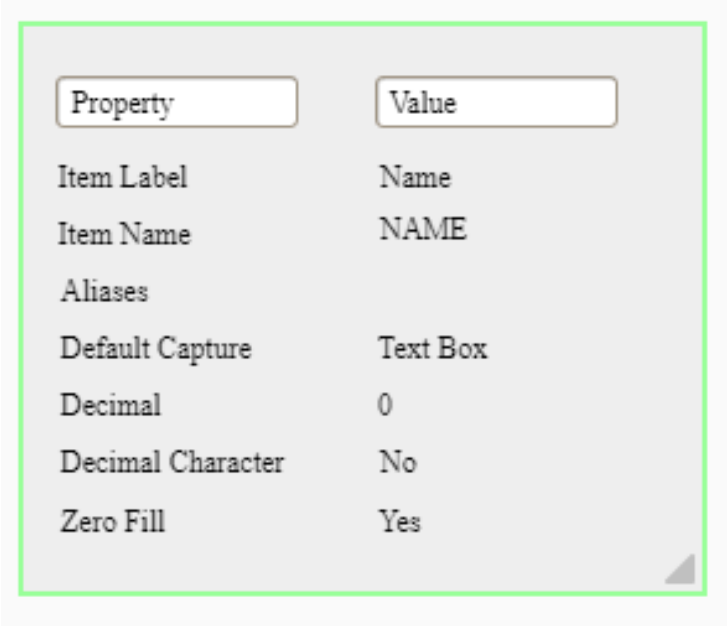


Figure 9 – Close-up view of the "extended" variables in the grid.

Finally, a green circle in the bottom right-hand side of the interface allows for adding new variables to the end of the dictionary record. See figure 10.

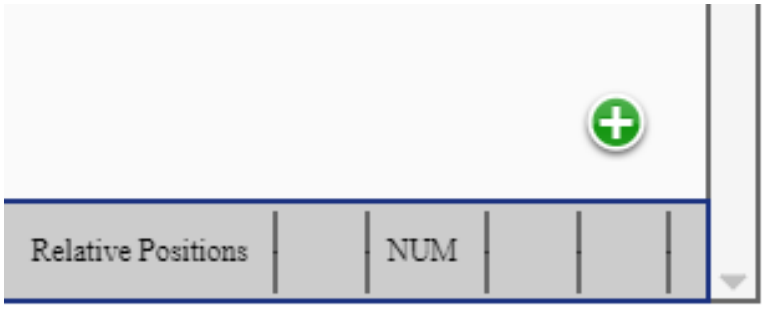


Figure 10 – Close-up view of add variable button.

4.2 Tree selector with additional indicators

A secondary effort of the redesign was to make it more obvious which tree tab was selected. The shading of the tab itself was made more prominent. Refer back to figure 4 for a comparison. Each tab includes an updated icon. The active tree's panel is now colored to match the updated icon. See figure 11, 12, and 13 to see the results of the design for each case tree.

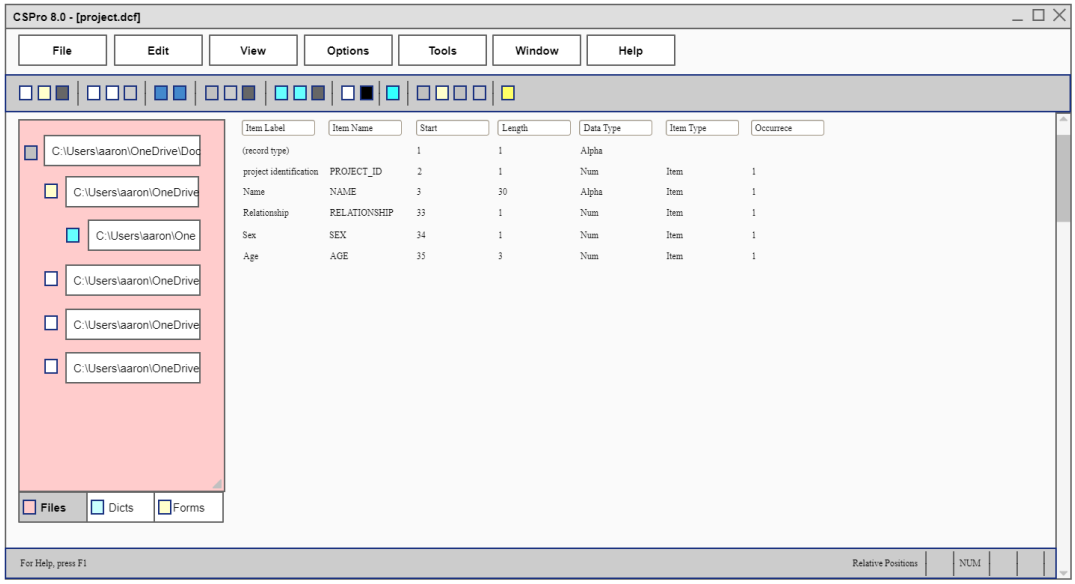


Figure 11 – File tree selected in the dictionary editor.

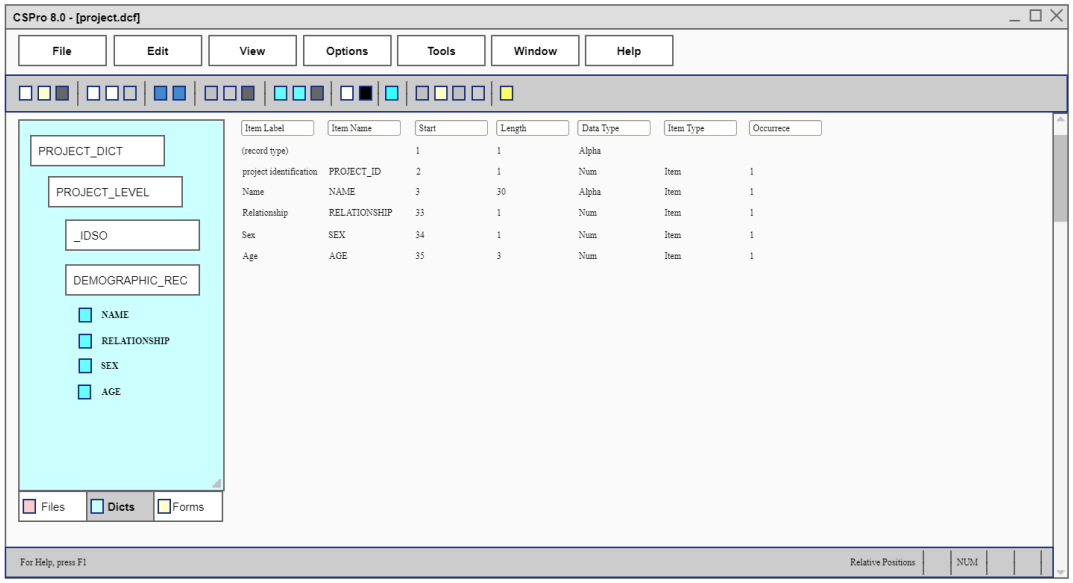


Figure 12 – Dictionary tree selected in the dictionary editor.

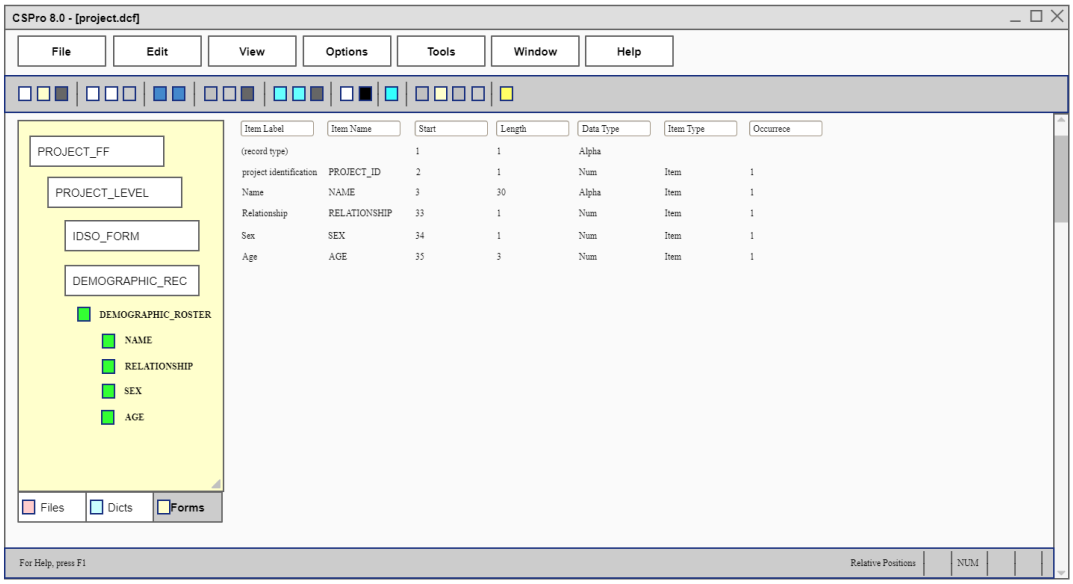


Figure 13 – Form tree selected in the dictionary editor.

5 INTERFACE JUSTIFICATION

5.1 Preserving the positive elements

All three participants during the interview praised the dictionary editor on how it represented its contents. The dictionary editor succinctly represents a lot of content while allowing the user to efficiently navigate it all.

The tree selector allows the user to view elements of the files, dictionaries, and forms while still being able to view the dictionary editor. If floating windows were used (e.g., GNU Image Manipulation Program) two-thirds of the screen would be used to display this information. I will compare this design to Norman's concept of *physical constraints*. The CSPro designers realized the file, dictionary, and form trees were beneficial to have available while working in the dictionary editor, but also realized the user never needed to access the file, dictionary, or form trees at the same time. By creating the tree selector in this way, the designers constrained the user to never allow them to unnecessarily clutter their screen.

Each individual variable and its attributes are organized into a single unit, which allows for the user to also "pull back" and view many variables at a glance. Allowing the user to view a variable relative to its neighbors helps the user bridge the *gulf of execution*, because this representation is conceptually similar to the questionnaire the user has already designed. Here the *design principle of structure* organizes the dictionary editor to match the *mental model* of the questionnaire the user is creating.

When a variable is added to the dictionary, the dictionary editor uses a series of smart defaults to reduce the amount attributes that need to be filled out. For instance, if the user globally sets the option to display the decimal character in the data file this attribute is remembered and each new variable added will have this attribute prefilled. Here the *design principle of ease* is used to reduce the fatigue of adding hundreds or thousands of variables.

CSPro ties many of its concepts to real-world conventions. For instance, the dictionary editor is where a user defines all the variables for a questionnaire. In that way it is similar to how a dictionary defines all the words in a language. Throughout CSPro the dictionary editor is represented with a blue book. Here the *design principle of mapping* is used to associate the concept of a physical dictionary which the user understands with the new concept of a dictionary editor.

CSPro has different components that can be associated with a variable. For instance, a user can write logic to affect how a variable behaves in a questionnaire or a user can add questionnaire text for the variable. Each of these tasks are done outside of the dictionary editor. However, the dictionary editor allows you to delete a variable which affects these components. When a dictionary variable is

deleted a choice was made to preserve the logic and question text. This allows for a variable to be re-associated with the logic or question text, if it is added back. Here the *design principle of tolerance* allows the user to avoid this domino effect where deleting a dictionary variable could also delete associated logic and question text by allowing it to be undone.

5.2 Addressing criticisms

The most significant issue identified is users sometimes edit the dictionary outside of the dictionary editor. That is a clear indicator that there is a problem, because that is the very purpose of the dictionary editor. Also, by editing the dictionary outside of the dictionary editor the validation done within CSPro is sidestepped which can lead to users having to discard their work. For small to medium changes the dictionary editor is preferred. It is only when the changes are large that it becomes worthwhile to work in an outside program like Excel. The reason for this is there is a learning curve and overhead to creating and editing the variables outside of the dictionary editor.

To address this, I looked closely at what makes the task of editing variables in Excel better. CSPro represents variable data in a series of rows comprised of cells, so it is easy to see how Excel could hold the same information. See figure 14 as an example of two variables being displayed within two rows and five cells each.

Name	NAME	3	30	Alpha
Relationship	RELATIONSHIP	33	1	Num

Figure 14 – Two variables and their attributes organized into rows in the dictionary editor redesign.

However, there are a number of small things Excel makes easier. Moving around with the arrows keys is *consistent* within Excel while in the dictionary editor the behavior will change depending on the type of cell selected. Further, the dictionary editor only allows selection of an entire row. Which means navigating the individual cells relies entire on mouse clicks. The redesign changes this and allows individual cells to be selected, see the green selection box in figure 14. Once a cell in Excel is selected it is editable. In the dictionary editor the cell must be "clicked into." This can become tedious and slow down an expert user. Further, this is a source of *action-based slips*. If the user clicks to select a cell and then again clicks to "click into" the cell too quickly CSPro will detect a double-click and

open the value set menu. While this behavior can be disorienting it also lacks *consistency* between applications. When double-clicking on a cell in Excel the cursor is placed in the cell. Because of the prevalence of Excel, CSPro should mimic Excel's behavior when it comes to rows and cells whenever possible. Finally, adding the search and replace functionality allows the user to locate variables quicker and make large number of changes faster.

The number of attributes a variable can have often makes the row extend off the screen horizontally. This tends to hide information and negatively impact the *perceptibility* of the system. The redesign moves the attributes that appear at the end of the rows and displays them vertically in a grid. The grid replaces what would have been cells running off the screen, and creates a better user experience. The cells towards the end of the row can be set on a global level. For example, all real numbers display the decimal point in the data file. While these attributes need to be accessible, they only need to be accessible occasionally. From a birds-eye-view the most important information is visible and the rest of the attributes are visible when the variable is selected.

To add a variable the user needs to right-click and select "add item" from a context menu. The redesign moves this action out of the context menu and to the bottom-right of the dictionary editor as a floating button. While it improves the *discoverability* of the interface, it also optimizes the number of actions to complete a common task. Additionally, the accelerator CTRL+A is associated with the new button. By providing a button for novice users and an accelerator for expert users the interface is utilizing the *design principle of flexibility.*

While the tree selection panel is a very effective use of space it is a source of mistakes for novice users and slips for expert users. Often a novice user will not entirely understand the difference between the dictionary and form trees and perform a *knowledge-based mistake*. While expert users will understand the difference between the three trees, they look similar enough that the users will perform the right task on the wrong object leading to an *action-based slip*. Each tab includes an icon that *maps* the dictionary editor task to a real-world concept. For example, the three icons used are a file, dictionary book, and a form document. Additionally, the color of the icon matches the color fill of the panel. This is meant to be a subtly, but *perceptible* reminder of which panel is selected. However, the user

does not need to rely on the color. The more prominent shading of the selected tab is the primary indicator of which tree panel is shown.

6 EVALUATION PLAN

The redesigned dictionary editor incorporated multiple smaller changes in an attempt to improve the efficiency of adding variables to the dictionary editor. If a functional prototype was available, I would conduct an empirical evaluation. However, I would not expect to have a functional prototype at this stage of the design lifecycle. Instead, I will conduct a qualitative evaluation with the existing wireframe prototype.

To evaluate the interface, I will have participants perform a *think-aloud protocol*. The think-aloud protocol will allow me to observe the participant and listen to their thoughts as they use the prototype. This allows me to "get inside their head" and better understand their actions. However, this will require *synchronous* session where I am able to meet each participant in person. I regularly perform workshops on CSPro, so I will recruit participants from these workshops. This will guarantee the participant is a current user and is familiar with the current dictionary editor. I will meet with each participant individually and only show them a single prototype. I will conclude the evaluation with a brief interview to gather qualitative comparisons between the existing and redesigned interfaces.

I will do the following to avoid biasing the evaluations. To mitigate *recall bias*, I will record the think-aloud protocol. The advantages to recording the think-aloud protocol is it is automated and comprehensive. Afterwards, I can refer back to the recording if there is an issue of recall. To lessen *social-desirability* I will limit the background information I give. For instance, I will not tell the participant the motivation behind the redesign which is to reduce the participants reliance on Excel to change large number of variables. To avoid *observer bias* I will script the think-aloud instructions. To reduce *confirmation bias* I will have a colleague review my interview questions for leading questions. Finally, to diminish *voluntary-response bias* I will view the information gathered from this evaluation as needing to be supported by further data gathered in the design lifecycle process.

6.1 Scripted instructions for think-aloud protocol and interview questions

Hi [insert participant's name], thank you for meeting with me today. I am evaluating a redesigned CSPro interface. Specifically, the redesign is focused on the dictionary editor. I would like you to take a look at this wireframe prototype. The new elements you will notice are:

1. Each cell can be selected, notice the green box around the variable NAME.
2. Once a cell is selected the cell selection can be moved by pressing the arrow keys or by using the mouse.
3. As soon as a cell is selected it can be edited by typing on the keyboard.
4. Notice that when a variable is selected a grid appears to display additional attributes.
5. When a cell is selected the search bar appears in the top-right corner.
6. The search functionality will search all cells within each variable.
7. The search bar can be expanded to facilitate search and replace functionality.
8. The search and replace bar will search all text cells and allow the replacement of a substring.
9. In the bottom-right corner a floating button was added to add variables.
10. The add variable functionality can also be activated with the accelerator CTRL+A.
11. Notice that the tree selector tabs are shaded more prominently when selected and the fill color of the panel matches the icon in the tab.

Take a minute and review the wireframe prototype and let me know when you are ready to proceed. (pause)

Next, while looking at the wireframe prototype can you walkthrough how you would add a variable? While walking through the process please say what action you are taking and why. (pause)

Thank you, for walking through the process with me. I would like to ask you a few additional follow-up questions.

1. Do you feel the prototyped interface made it more or less efficient to add variables in the dictionary editor?
2. Typically, when starting a new survey with over 100 questions do you use the dictionary editor to add the variables?

3. (if another program) What program do you use to add the variables?
4. (if another program) Do you believe the prototyped interface has addressed the reasons you use another program?
5. (if not) What changes could be made to the prototyped interface, so it was not necessary to use another program?

That concludes the evaluation. Again, thank you so much for your valuable feedback.

7 APPENDICES

7.1 Forum review

New to CSPro; trying to convert paper survey to data

- 24 posts
- Issue: Used subitems when multiply occurring field should have been used.
- Issue: Did not understand why some functionality was grayed out.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=4508&p=14099&hilit=dictionary+editor#p14099

Data Viewer 7.0 Logic helper

- 5 posts
- Issue: Distinguish between two value sets with same label.
- https://www.csprousers.org/forum/viewtopic.php?f=4&t=3968&p=12303&hilit=dictionary+editor#p12303

Using Local Font in Dictionary and Form

- 7 posts
- Issue: Pasting text from the Oriya language directly into the dictionary.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=3006&p=9646&hilit=dictionary+editor#p9646

Dictionary with record length that does agree with the variables

- 5 posts

- Issue: Editor dictionary in outside editor, because they hoped it would be faster.
- https://www.csprousers.org/forum/viewtopic.php?f=7&t=2743&p=9058&hilit=dictionary+editor#p9058

Help with Error (W 10102) – "…index could not be created…"

- 3 posts
- Issue: Confusion on association between dictionary and data file.
- https://www.csprousers.org/forum/viewtopic.php?f=10&t=2415&p=7903&hilit=dictionary+editor#p7903

Check is it falls in a certain range

- 4 posts
- Issue: How to check if response is in non-continuous value set.
- https://www.csprousers.org/forum/viewtopic.php?f=8&t=2049&p=6686&hilit=dictionary+editor#p6686

Dropbox Sync

- 42 posts
- Issue: syncdata requires dictionary name which can be found in dictionary editor.
- https://www.csprousers.org/forum/viewtopic.php?f=11&t=1925&p=6071&hilit=dictionary+editor#p6071

Delete value set label

- 4 posts
- Issue: User is looking for an automated way to delete all value sets in a dictionary.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=1816&p=5548&hilit=dictionary+editor#p5548

Excel to CSPro – Dictionary Macro

- 6 posts
- Issue: User has thousands of variables to create. Looking for faster way to add items and value sets to dictionary.

- https://www.csprousers.org/forum/viewtopic.php?f=1&t=1763&p=5300&hilit=dictionary+editor#p5300

How to Labeling Dictionary Record with Occ

- 3 posts
- Issue: Question on creating occurrence labels. User did not realize that they could right click on variable in dictionary editor and a context menu would appear.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=1702&p=5013&hilit=dictionary+editor#p5013

Do Cspro an open source project

- 6 posts
- Unrelated: Inquiry on open source version of CSPro.
- https://www.csprousers.org/forum/viewtopic.php?f=6&t=393&p=4790&hilit=dictionary+editor#p4790

Sub-Items

- 13 posts
- Issue: Subitem included overlapping variables causing an error that the variable cannot be dropped on the form.
- https://www.csprousers.org/forum/viewtopic.php?f=7&t=1468&p=4235&hilit=dictionary+editor#p4235

Bengali – CSPro Language at install

- 2 posts
- Issue: Supporting additional languages in CSPro.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=1395&p=4043&hilit=dictionary+editor#p4043

Roster with fixed title

- 4 posts
- Issue: Unfamiliar with occurrence labels.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=882&p=2236&hilit=dictionary+editor#p2236

Encounter an improper argument

- 8 posts
- Issue: Bug with the dictionary editor displaying errors.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=876&p=2203&hilit=dictionary+editor#p2203

Create relation

- 2 posts
- Issue: Create relation between two rosters.
- https://www.csprousers.org/forum/viewtopic.php?f=1&t=857&p=2152&hilit=dictionary+editor#p2152

7.2 Interview questions

- How long have you been working with CSPro?

Instructions: *For reference open the dictionary editor in CSPro*

- Name 1 to 3 things the dictionary editor does well?
    - How / why does the dictionary editor do this well?
- Name 1 to 3 things the dictionary does poorly?
    - How / why does the dictionary do this poorly?

Prompt: *Users sometimes circumvent the dictionary editor and edit the dictionary directly in an attempt to speed up the process of adding hundreds of variables and value sets. Which can lead to ill-formed dictionaries.*

- Do you have any suggestions on how the CSPro interface could be modified to better facilitate entering many variables / value sets?
- Do you think this is something best done outside CSPro?

Prompt*: New users often use subitems, because they are intuitive. However, we often recommend not using them.*

- Do you use subitems? If so, when do you use them?

7.3 Interview responses

Participant 1

Name 1 – 3 things the dictionary editor does well

- View variable names
- View variable types

- Organizes underlying data in similar fashion to the paper questionnaire

Name 1 – 3 things the dictionary editor does poorly

- Fix length fields
    - Users who edit variables while collecting data can create a huge mess and nothing warns them
- Not allowing aliases for variables
    - Allows for the reuse of logic

UI changes to better facilitate entering many variables / value sets

- Allow search and replace on all variables / value sets

Do you use subitems? If so, how?

- Yes, uses them for dates and occupation codes. Additionally, could use for fertility to section to check if it is blank across multiple fields (e.g., males).
- Issue: he subitem is a symbolic grouping the lies above the data. If data gets shifted around or subitem is set wrong the symbolic grouping will be wrong, but the application developer won't know.
- They're convenient, but most times not worth the risk

Participant 2

Name 1 – 3 things the dictionary editor does well

- Visual / representation
    - Three displays with dictionary tree, layout, and content
- Succinctly display a lot of information
- Consistent w/ the way the controls work
    - In ISSA (predecessor to CSPro) conceptually similar functionality had different hotkeys in different contexts

Name 1 – 3 things the dictionary editor does poorly

- Creating subitems from items will move children
- Up / down arrow navigation triggers dropdown when in the data type cell
- Selecting item + another click in cell to edit field sometimes registers as a double click if done to fast and takes you to the value set.

UI changes to better facilitate entering many variables / value sets

- Excel is commonly used to design value sets. Improve functionality and allow import from other databases in the census/survey ecosystem

Do you use subitems? If so, how?

- Yes, DOB and industry/occupation
- Facilitates tabulation at different granularities
- Math of dates can be difficult

Participant 3

Name 1 – 3 things the dictionary editor does well

- Attributes have default values when adding a new item.
- Layout view allows for a quick overview of data.

Name 1 – 3 things the dictionary editor does poorly

- Functionality hidden in menu. For instance, you have to know to look for occurrence labels.
- Record don't allow for grouping of items. For instance, fertility or education.
- Records don't do as good a job of as forms at making additional information apparent at a glance like the number of occurrences or required constraint.
- Max hierarchy for dictionary is four, but six would be ideal (dictionary, level, record, sub-record, item, subitem)
- Hide items that are meta data (e.g., add more persons). Also, allow value sets to be hidden depending on context (tabulation, batch, data entry).
- Row layout can extend off screen. Column widths do not resize bases on resolution.
- Find functionality could include "find all references" like Visual Studio.

UI changes to better facilitate entering many variables / value sets

- Dictionary macros does help with this. Generate items and copy + paste to Excel and copy back.
- Allow importing of additional formats.

- Allow free form editing. Currently the dictionary does validation after entering the data. Excel allows the free to move around and leave fields blank until ready to enter data.
- Replace all

Do you use subitems? If so, how?

- When engine is rewritten, I will use them. Record, item, subitem allows for 3 dimensions, but CSPro only supports 2 dimensions. Forms will stop you from entering 3 dimensions, but nothing will stop user if they directly edit dictionary.
- Very useful for industry / occupation.
- Granularity when doing tabulation.
- Convenient to get a 1st or 3rd digit.
- Some functions require 8-digit date. Subitems are ideal here.
- Subitems are flexible and serve many purposes, but no specific use for subitems. More times than not they are used for a creative workaround and a new feature would be better suited.