CS-TR-3692 Sept. 1996

Putting Visualization to Work: ProgramFinder for Youth Placement

Jason Ellis, Anne Rose, Catherine Plaisant

Human-Computer Interaction Laboratory
Institute for Advanced Computer Studies
University of Maryland, College Park, MD 20742-3255
http://www/cs.umd.edu/projects/hcil/
{jellis, rose, plaisant}@cs.umd.edu

Abstract

The Human-Computer Interaction Laboratory (HCIL) and the Maryland Department of Juvenile Justice (DJJ) have been working together to develop the ProgramFinder, a tool for choosing programs for a troubled youth from drug rehabilitation centers to secure residential facilities. The seemingly straightforward journey of the ProgramFinder from an existing user interface technique to a product design required the development of five different prototypes which involved user interface design, prototype implementation, and selecting search criterion. While HCIL's effort focused primarily on design and implementation, DJJ's attribute selection process was the most time consuming and difficult task. We also found that a direct link to DJJ's workflow was needed in the prototypes to generate the necessary "buy-in". This paper analyzes the interaction between the efforts of HCIL and DJJ and the amount of "buy-in" by DJJ staff and management. Lesson learned are presented for developers.

Putting Visualization to Work: ProgramFinder for Youth Placement

Jason Ellis, Anne Rose, Catherine Plaisant

Human-Computer Interaction Laboratory
University of Maryland Institute for Advanced Computer Studies
College Park, MD 20742-3255
{jellis, rose, plaisant}@cs.umd.edu
http://www.cs.umd.edu/projects/hcil/

ABSTRACT

The Human-Computer Interaction Laboratory (HCIL) and the Maryland Department of Juvenile Justice (DJJ) have been working together to develop the ProgramFinder, a tool for choosing programs for a troubled youth from drug rehabilitation centers to secure residential facilities. The seemingly straightforward journey of the ProgramFinder from an existing user interface technique to a product design required the development of five different prototypes which involved user interface design, prototype implementation, and selecting search criterion. While HCIL's effort focused primarily on design and implementation, DJJ's attribute selection process was the most time consuming and difficult task. We also found that a direct link to DJJ's workflow was needed in the prototypes to generate the necessary "buy-in". This paper analyzes the interaction between the efforts of HCIL and DJJ and the amount of "buy-in" by DJJ staff and management. Lesson learned are presented for developers.

KEYWORDS: technology transfer, visualization, dynamic query, legal systems, matching

INTRODUCTION

For the past two years, the Human-Computer Interaction Laboratory (HCIL) has been working with the Maryland Department of Juvenile Justice (DJJ) to redesign the user interface of their information system which is used to process approximately 50,000 juvenile complaints per year. The first year consisted of performing 22 field visits, administering the Questionnaire for User Interaction Satisfaction (QUIS) to 332 DJJ personnel, and making short and long term user interface recommendations (Rose, Shneiderman & Plaisant, 1995; Slaughter, Norman & Shneiderman, 1995). In the second year, we continued with extensive prototyping with an emphasis on supporting DJJ's workflow related to youth case management.

One case management function involves placing youths in a variety of programs that meet their individual needs, ranging from community-based drug treatment programs to secure residential facilities. Currently, DJJ chooses from about 250 programs. This process involves searching through a 4-inch manual to find the best program. Not only is very time consuming but there is also the potential bias of choosing the first program found, as opposed to the one best suited to the needs of the youth. It was immediately obvious to us that HCIL's earlier dynamic query (DQ) research could be applied here since they had been designed to solve problems before. The ProgramFinder was designed to allow DJJ to quickly and easily select the best program(s) for a youth from among all the programs matching the set criterion.

BUILDING ON EXISTING TECHNIQUES

One of the original dynamic query prototypes was the HomeFinder, a tool for browsing homes for sale in an area (Figure 1). Dynamic query (DQ) applications support fast and easy exploration of data by allowing users to make queries by adjusting sliders and selecting buttons while the search results are continuously updated in a visual display (e.g., x/y scatterplot, map, etc.) (Williamson & Shneiderman, 1992; Ahlberg & Shneiderman, 1993). Instead of the HomeFinder's map of Washington, D.C, the ProgramFinder plots the available programs on a map of Maryland (Figure 2). Adjusting the placement controls updates the display which shows a dot for each program that matches. A click on a program provides more details and the press of a button generates the appropriate paperwork.

This paper describes the seemingly straightforward conversion of the ProgramFinder from a research prototype to a real product, and analyzes the interaction between the efforts of HCIL and DJJ and the amount of "buy-in" of DJJ staff and management (i.e., how excited they seemed to be about the prototype). We found that many levels of prototyping were still needed (5) and that the choice of the search criterion was the most time consuming (and the most conflict generating) task. A direct link to the workflow was also needed in the prototypes to generate the necessary "buy-in".

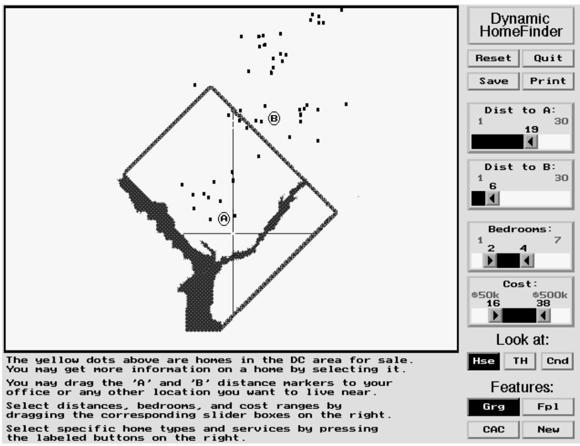

Figure 1. Original HomeFinder research prototype

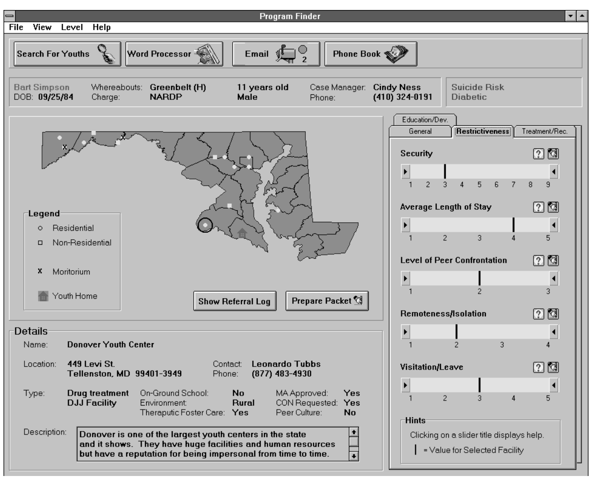

Figure 2. Final ProgramFinder prototype

DESIGN PROCESS

The process of evolving the ProgramFinder design from the original HomeFinder concept (Figure 1) to the final design (Figure 2) involved five different prototypes:

- IVEE prototype
- Initial Customization prototype
- Comparison prototype
- Testing prototype
- Final prototype

The primary effort involved in developing each prototype consisted of customizing the user interface design, implementing the prototype, and deciding on the search criterion. The level of effort in each of these categories varied significantly by prototype and so did the amount of user "buy-in".

The initial IVEE prototype was developed in a few hours to illustrate the ProgramFinder concept to DJJ. With DJJ's go ahead to continue, the Initial Customization prototype was then developed. This is when development started to focus more intently on DJJ's workflow and as a result DJJ's "buy-in" increased dramatically. DJJ also began working harder on choosing the selection criterion. It became obvious that staff had vastly different opinions than management. A comparison prototype was developed to illustrate the worker's ideas to management. After considerable debate, management decided on a set of criterion to use and a testing prototype was developed so preliminary usability testing could be performed. After the attributes were selected, the design effort increased because DJJ started to really react to all the details in the prototype and requested many modifications. The testing prototype required increased implementation effort because there was little working functionality in the previous prototypes.

IVEE Prototype

The first prototype (Figure 3) was built in a few hours using the Information Visualization and Exploration Environment (IVEE) (Ahlberg & Wistrand, 1995). IVEE automatically creates DQ interfaces for given datasets. The dataset used to generate the ProgramFinder was entirely mocked up by HCIL.

The major drawback of using IVEE was that it ran on Sun workstations and DJJ only uses PCs. For demonstration purposes, we resorted to using a slide show of IVEE screens, in conjunction with a live demo of the HomeFinder to show the smooth DQ interaction. DJJ's initial reactions were positive and they asked us to continue.

Initial Customization Prototype

The implementation and attribute selection efforts increased during this phase. DJJ started to get more involved. They provided us with more detailed information about their placement process and proposed a set of attributes. These attributes allowed users to specify the "best" value (i.e., the ideal value for the youth) within a range of values. This had the advantage of allowing the selected programs to be rank ordered. However, this required a few modifications to the standard range slider behavior:

- <u>Best Values</u>: The range sliders were modified so a click on a value underneath the slider selected that value as the "best" one for that attribute and

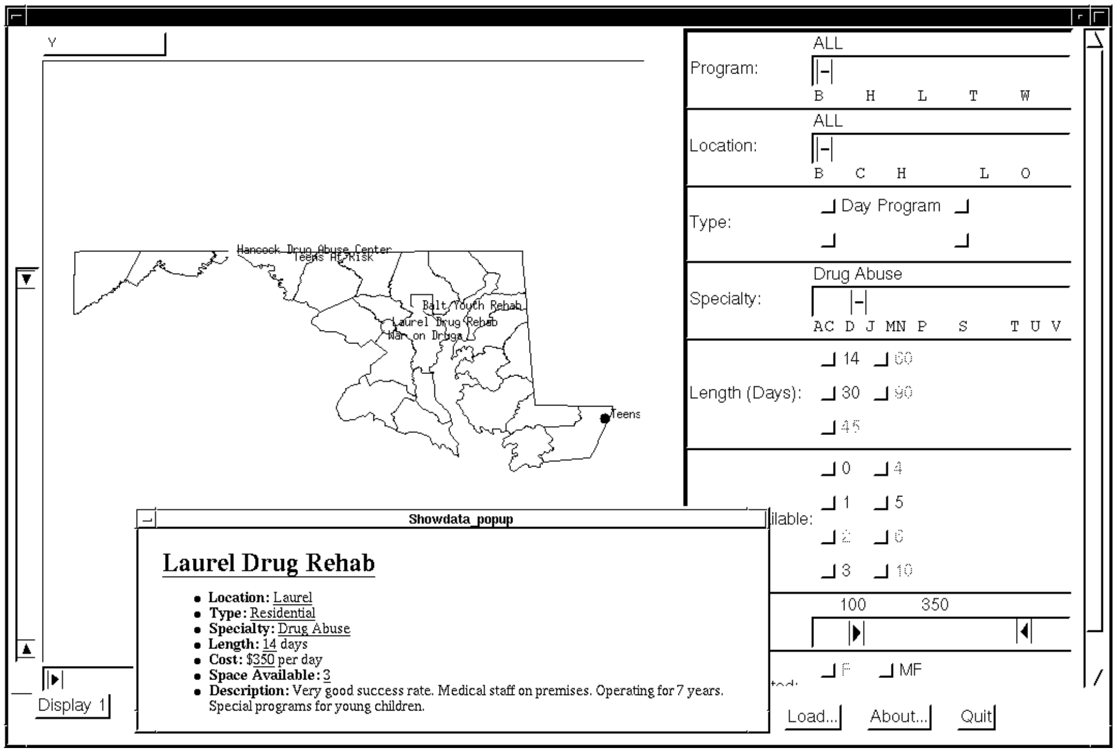

Figure 3. IVEE prototype

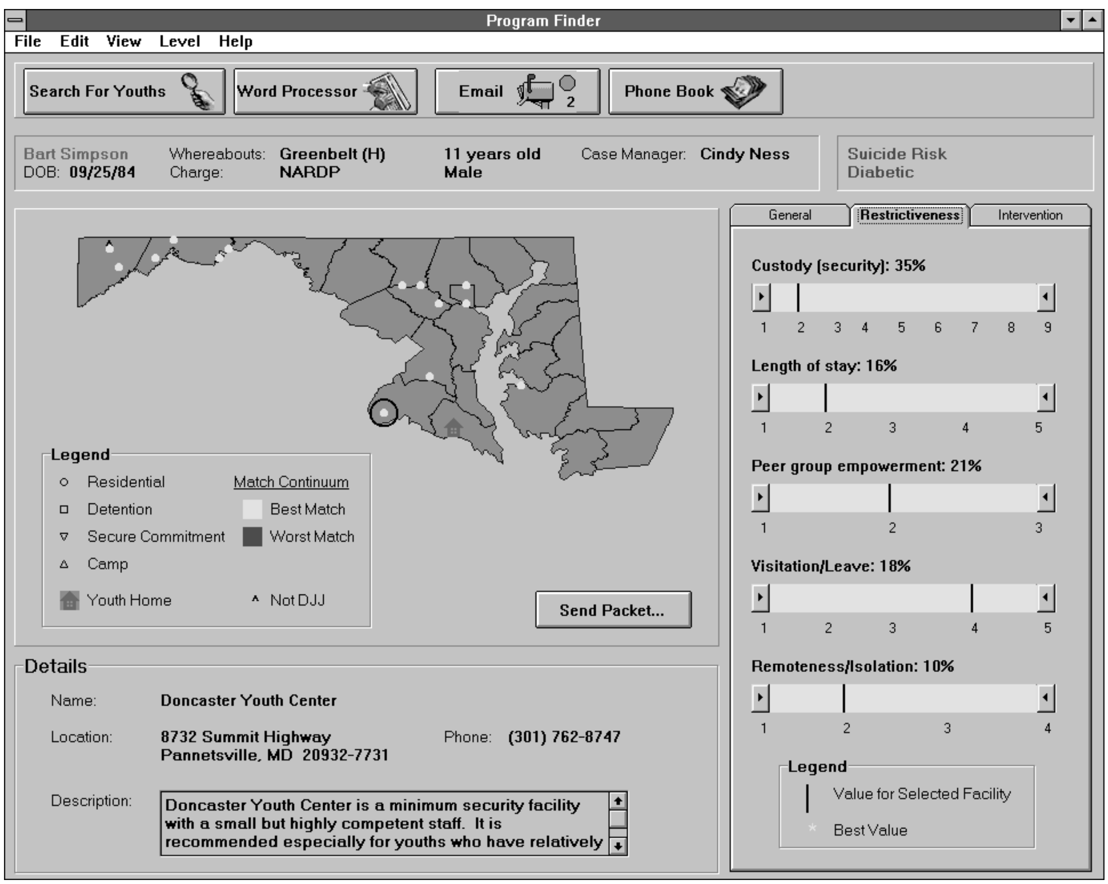

Figure 4. First Delphi prototype

marked it with a yellow star. The matching programs are then color-coded from bright yellow (for best) to dark red (for worst) with respect to how close they are to the best values.

- Current Values: Vertical black lines were also added to the range sliders to show the actual value of the selected program.

The second prototype was developed using Borland's Delphi on PCs (Figure 4). This gave us the advantage of being able to run on DJJ's machines plus IVEE did not allow us to design a more customized interface for DJJ. A "Send Packet…" button was added to demonstrate how users could select a program and then automatically generate the required paperwork.

DJJ's "buy-in" jumped dramatically when they were shown the prototype. They were very excited to see a DJJ document popup when the "Send Packet …" button was pressed. They immediately started discussing how they could market it to other juvenile justice agencies. They also started envisioning other ways the ProgramFinder could be used. For example, they suggested adding a referral log so the acceptance and rejection patterns of different programs could be monitored. Having a strict set of criteria for what a program provides would allow DJJ to hold the programs to that standard. DJJ was quickly moving away from a casual exploratory effort to a more serious product design effort. Ironically, the ProgramFinder was not even a tool DJJ anticipated needing initially. Now there were discussions about how it could be used to explore programs than currently possible. By coordinating with other MD agencies, they could choose from over 2000 programs. We believe this shift was primarily due to the customization effort, even though it was relatively simple.

Comparison Prototype

Shifting toward a more serious design effort, we started working more closely with the DJJ staff and discussing their needs in detail. Up to this point, our discussions had been primarily with middle and upper management. While the users were pleased with the general concept of the ProgramFinder, they were concerned that the proposed attributes were too limiting and did not correspond to how they currently did their jobs. The comparison prototype was developed to illustrate their proposed changes to management (Figure 5).

The major effort involved with this prototype was deciding on the new data attributes which were significantly different from the attributes proposed by management. Another difference was how to specify the value of an attribute. Instead of specifying a range of values, the workers wanted to rank each value on a scale from not important to required. This required the design of an attribute ranker widget. Some other minor changes included reducing the number of program types in the legend, adding more fields to the details area, and creating a "Show Referral Log" button.

There were now two significantly different prototypes, in terms of attributes, that needed to be brought to some consensus. Management was presented with both prototypes and the strengths and weaknesses of each were discussed. After a month of deliberation, management chose the Initial Customization prototype. The rationale was that the checklist attributes in the Comparison prototype did not engage the user in the selection process as much as the range sliders. There was also the concern that users might ignore critical areas in the lengthy checklists which would greatly effect the level of service a youth receives. DJJ decided that it was preferable to provide a few attributes with broad implications and ask users to consider all of them. The decision not to use the worker's criterion (Comparison prototype) decreased the their "buy-in" temporarily. While the ProgramFinder would still help the staff perform their jobs, it had also become the vehicle by which their jobs were being redefined.

Testing Prototype

Management's decision to use the original data attributes spawned additional discussions about the set of attributes to be included in the testing prototype, which was used to perform initial usability testing. Interestingly, after working with management's attributes for a few hours, the workers decided they were sufficient after all. However, they did propose a few new attributes to include in the testing prototype.

Management requested that the color coding not be included in the testing prototype because they felt it would unduly bias the selection process. The concern was that workers might just select the highest ranked program (e.g., the one with the "best" color) and not take into account other suitable programs. DJJ wanted to avoid creating a tool that gives the "perfect answer." They wanted the ProgramFinder to narrow down the number of programs and then require the workers to examine each of the remaining programs in-depth to select the best one.

HCIL's major effort in developing the testing prototype (Figure 6) was implementation since there was still very little working functionality in the previous

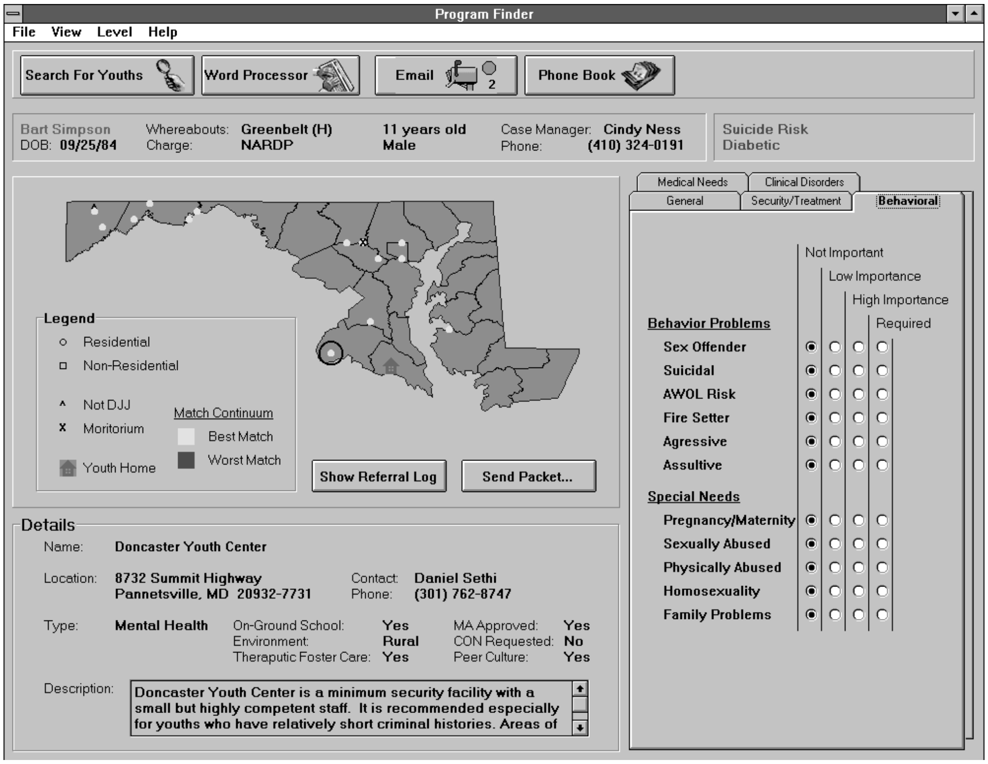

Figure 5. Comparison prototype

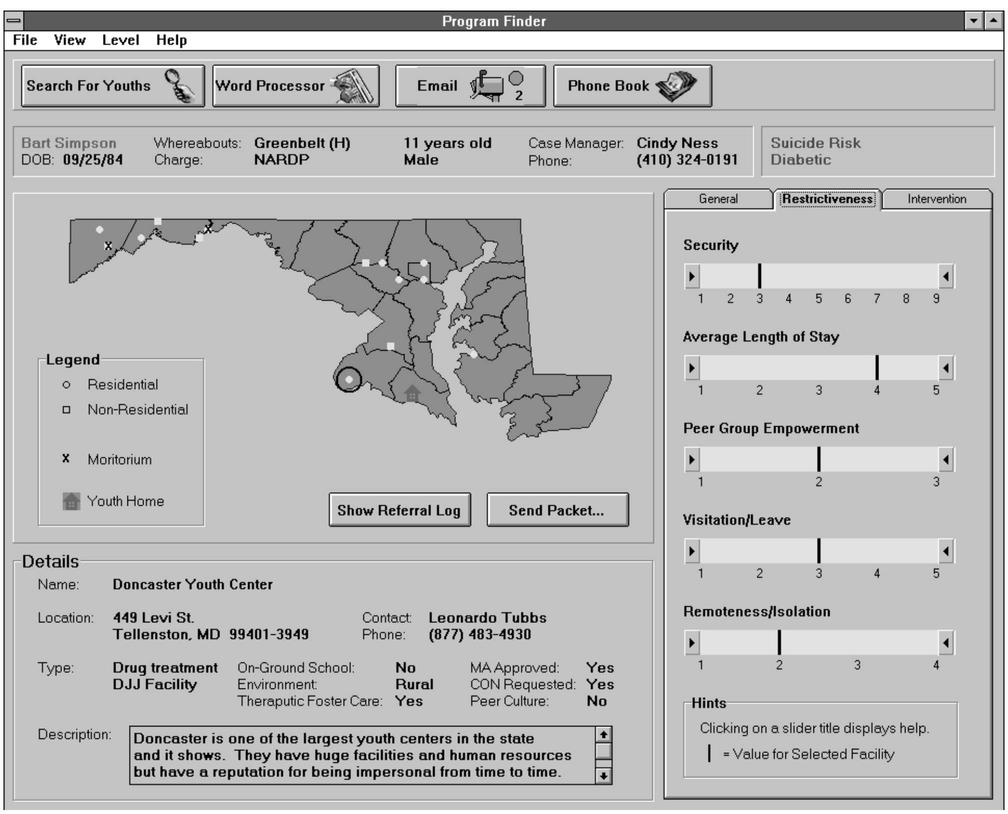

Figure 6. Testing prototype

prototypes. The slider implementation in particular required the most time. Similar controls are available in the public domain but none had all the functionality DJJ needed.

Final Prototype
During testing, several changes were proposed to the prototype. The result was the final prototype design (Figure 2).

INFORMAL USABILITY TESTING

Preliminary usability testing was conducted on the testing prototype. The goal was to give users hands-on experience while HCIL gained valuable design feedback. The testing consisted of two sessions with a total of seven users, a limited but representative group of users. Each session was divided into four sections: training, testing with representative tasks, filling out a questionnaire, and discussion. Screen mockups illustrating solutions to problems discovered during the first session were presented to users in the second session for their feedback.

Users' reaction was very positive. This was not surprising since they had been involved in the design from the early stages. However, several usability issues did emerge during testing that were incorporated into the final design (Figure 2).

1- Addition of Textual Display - Users noted that plotting the programs on a map was not very useful since the location of a program is not normally taken into account when placing a youth. A textual display was added (showing a list instead of a map) as a more effective way to review the few best matches once the filtering was done.

2- Reinstate the "Best" Values - DJJ reversed its decision about color coding with respect to "best" values. Although they were initially concerned that ranking programs might bias the selection process, after using the system they realized the color coding could assist workers when there is no program that matches a youth's needs fully. In addition, assigning "best" values would provide a clearer picture of what sorts of programs are needed.

3- More Integrated Help - Users in the first session recommended allowing selections from the help facility which provided more detailed information about the slider values. A sample screen illustrating how users could select the maximum and minimum values via

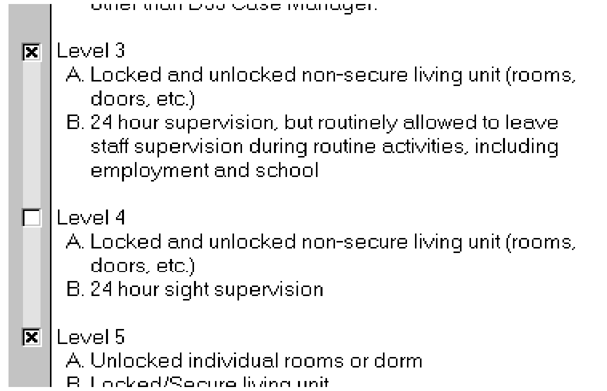

Figure 7. Help facility supporting range selection

check boxes in the help facility was presented to the users in the second session (Figure 7). The new interface met with mild approval but the users felt the range sliders would be more convenient to use once they learned how to use them.

4- Attaching Notes - while the workers were using the ProgramFinder, they found that they wanted to record comments about their settings. A small icon above each slider was added that when clicked would display the portion of the placement packet related to that particular attribute.

5- Simplifying Range Sliders - several users expressed difficulty using the range sliders. They were frustrated using the sliders when they knew the exact range they wanted. One suggestion was to enhance the range sliders to allow users to select a range by dragging their mouse across the values shown below the slider. This would only require one action as opposed to the two drags required by the standard range slider.

6- Reordering Sliders - the order and categorization of the attributes was raised as an important issue. The decision was to present the controls by workflow and allow users to redisplay them alphabetically.

LESSONS LEARNED

After several months of effort and five different prototype designs, we learned several important (and sometimes surprising) lessons that could benefit future developers.

Search criterion selection can be difficult - We initially anticipated that it would be a simple task, but choosing the search attributes for this visualization required the highest level of effort and caused the most conflict inside of DJJ. The Comparison prototype was developed solely for the purpose of exploring alternative attributes.

Customization Increases "buy-in" - We were surprised how much DJJ's "buy-in" increased after the Initial Customization prototype was developed. To us, it was merely a re-implementation of the IVEE prototype for the PCs and the customization added was very minor (a few buttons and scanned forms) but it had a dramatic impact on DJJ's ability to understand how the ProgramFinder could help them and to start planning for novel uses.

Interface design can initiate changes in work processes - In the case of the ProgramFinder, management decided to use a set of attributes that will significantly change how their workers select programs.

Presentation of similar applications stimulates early interest - Even though it is less effective than building a customized prototype, showing "live" demos of similar systems (e.g., HomeFinder) helps focus user thinking and bootstrap management "buy-in".

Creating alternative designs helps engage users - Illustrating functional differences through the creation of several prototypes is a very powerful tool. Users who initially expressed no came forward with strong ideas once concrete choices were presented.

SUMMARY AND CONCLUSION

The level of effort to convert an existing interface technique into a product design is significant. The entire process of designing the ProgramFinder involved six months of effort and five different prototypes and there are still issues to resolve (Figure 8).

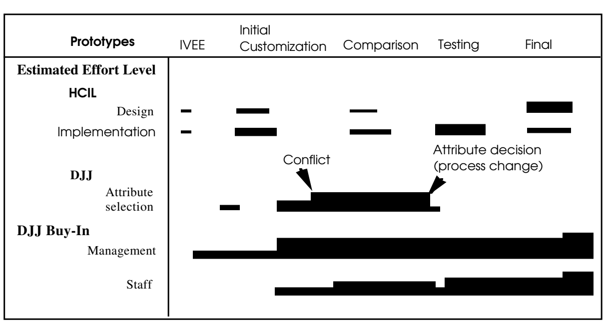

Figure 8. ProgramFinder Design Process. Line thickness indicates the relative amount of effort and "buy-in" and the length approximates the amount of time involved.

Selecting the search criterion was the most time consuming and conflict generating task. Demonstrating similar applications early on and adding custom workflow hooks to the prototypes increased "buy-in". Alternative designs were presented to increase user involvement. This effort also served as the catalyst for DJJ to redesign their work practice.

ACKNOWLEDGMENTS

We would like to thank Walt Wirshing and Dave Brimm from DJJ for their overall assistance. Additional thanks are due to all the DJJ personnel who took time out of their busy schedules to work with us. The preparation of this report was supported by funding from the Maryland Department of Juvenile Justice.

REFERENCES

Ahlberg, C., Shneiderman, B. (1993) "Visual Information Seeking: Tight coupling of dynamic query filters with starfield displays," *ACM CHI 94 Conference Proc. (Boston, MA, April 24-28, 1994)*, 313-317. Also appears in *Readings in Human-Computer Interaction: Toward the Year 2000*, Baeker, R.M., Gruden, J. , Buxton, W.A.S. & Greenberg, S., Eds., Morgan Kaufmann Pubs., Inc., (1995), 450-456.

Ahlberg, C., Wistrand, E. (1995) "IVEE: An Information Visualization & Exploration Environment," *Proceedings of IEEE Visualization '95 (Atlanta, October 1995)*, 66-73.

Rose, A., Shneiderman, B., Plaisant, C. (1995) "An applied ethnographic method for redesigning user interfaces," *ACM Proc. of DIS '95, Symposium on Designing Interactive Systems: Processes, Practices, Methods & Techniques (Ann Arbor, MI, Aug 23-25, 1995)*, 115-122.

Slaughter, L., Norman, K., Shneiderman, B., (1995), "Assessing Users' Subjective Satisfaction with the Information System for Youth Services (ISYS)," *Proceedings of Third Annual Mid-Atlantic Human Factors Conference (Blacksburg, VA, March 26-28, 1995)*, 164-170.

Williamson, C., Shneiderman, B. (1992) "The dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system," *Proceedings ACM SIGIR '92 (Copenhagen, June 21-24, 1992)*, 338-346. Also appears in *Sparks of Innovation in Human-Computer Interaction*, Shneiderman, B., Ed., Ablex (June 1993), 295-307.