Informing Automatic Generation of Remote Control Interfaces with Human Designs

Jeffrey Nichols • JeffreyN@cs.cmu.edu • Carnegie Mellon University • Human Computer Interaction Institute • http://www.cs.cmu.edu/~pebbles/pcu/

Analysis of Human Designs

We created interfaces for an AIWA CX-NMT70 shelf-stereo and an AT&T 1825 office telephone on a Microsoft PocketPC. We then analyzed these interfaces to understand the functional information that was needed for their creation. Our specification language includes the information that we found was needed to create remote control interfaces.

State Variables and Commands

Any manipulable appliance element can be represented by either a state variable or a command. Each state variable has a type that tells the interface generator how it can be manipulated. Some elements, such as the seek button on a radio, must be represented as commands, because their result cannot be described as a deterministic change to a variable. Today's remote controls could be generated by a specification that contained only commands.

Type Information

The type information of each state variable tells the interface generator how it can be manipulated. Boolean, enumerated, numeric, and string types are currently available.

Labeling Information

Each component must be labeled so that users can distinguish elements. Different form factors require different kinds of labels. Each semantic label for a variable or command in our specification language contains several strings that may be used based on available space.

Group Tree

Most interfaces can be described by a tree, where each leaf is a component and each branch is a panel. Our language uses a similar structure to suggest to the interface generator which components are found in a group, then those components will be placed in a row together.

Dependency Information

Enabling and disabling components is very useful for guiding users to available functionality. Our language contains formulas that specify when a variable or command is available based upon the values of other state variables. Equal-to, greater-than, and less-than relations are supported, and multiple relations may be combined with the logical AND and OR operations.

Future Work

We are currently working to extend our specification language and improve our automatically generated layouts.

Specification Language

• The specification language does not currently support lists, which are used by many appliances.

• Information must be added to the specification to support generating speech interfaces.

Interface Layout

• Our interface layout algorithms need some knowledge of standard component arrangements. We need better methods of including this information in our language.

• Our layout algorithm must deal appropriately with situations where there is not enough space. We are considering using backtracking, where previous decisions are re-thought in the context of newly discovered constraints.

Acknowledgements

This work was sponsored in part by an NSF grant, administered by the Institute for Software Integrated Systems, Pittsburgh, Pennsylvania. We thank Michael Smith, also at ISIS, for his continued support. We also thank Larry Fordwick, Brad Myers, Brad Myers, and Peter Lucas of MAYA Design. It was funded by grants from NSF, Microsoft, and the Pittsburgh Digital Greenhouse, and equipment and Peter Lucas of MAYA Design. It was funded by grants from NSF, Microsoft, and the Pittsburgh Digital Greenhouse, and equipment.