From Usage Scenarios to Widget Classes

Hermann Kaindl[1] and Rudolf Jezek[2]

[1] Siemens AG Österreich, PSE, Geusaugasse 17,
A-1030 Vienna, Austria
hermann.kaindl@siemens.at
[2] Siemens Business Services GmbH & Co, Erdberger Lände 26,
A-1030 Vienna, Austria
rudolf.j.jezek@sbs.at

In practice, designers often select user interface elements like widgets intuitively. So, important design decisions may never become conscious or explicit, and therefore also not traceable. We addressed the problem of systematically selecting widgets for a GUI that will be built from those building blocks.

Our approach is based upon *task analysis* and *scenario-based design*, assuming that envisaged usage scenarios of reasonable quality are already available. Starting from them, we propose a systematic process for selecting user interface elements (in the form of widgets) in a few explicitly defined steps. This process provides a seamless way of going from scenarios through (attached) subtask definitions and various task classifications and (de)compositions to widget classes. In this way, it makes an important part of user interface design more systematic and conscious.

More precisely, we propose to explicitly assign subtask descriptions to the interactions documented in such a scenario, to the steps of both users and the proposed system to be built. Through combining those subtasks that together make up an interaction, *interaction tasks* are identified. For these, the right granularity needs to be found, which may require task composition or decomposition. The resulting interaction tasks can be classified according to the kind of interaction they require. From this classification, it is possible to map the interaction tasks to a class hierarchy of widgets. Up to this point, our process description is seamless, while the subsequent selection of a concrete widget is not within the focus of this work.

For this mapping, we defined a hierarchy of widget classes according to a functional rather than the usual structural view of classifying widgets. This hierarchy covers the MS Windows style guide. We also defined a hierarchy of interaction tasks that reflects this widget hierarchy.

For an initial evaluation of the usefulness of this approach, we conducted a small experiment that compares the widgets of an industrial GUI that was developed as usual by experienced practitioners, with the outcome of an independent execution of the proposed process. Since the results of this experiment are encouraging, we suggest to investigate this approach further in real-world practice.

Although this systematic process still involves human judgement, we hope that it contributes to a better understanding of the relationship between usage scenarios and user interface elements through the various tasks involved. More generally, it should help to better understand a certain part of user interface design.

M. Reed Little and L. Nigay (Eds.): EHCI 2001, LNCS 2254, pp. 35–36, 2001.
© Springer-Verlag Berlin Heidelberg 2001

Discussion

G. Phillips: Is your work intended to support - automatic generation of user interfaces - support to allow inexperienced user interface designers to produce better user interfaces
R. Jezek: Automatic generation is a long term goal. For now, we are aiming for support to inexperienced designers. Experienced designers can usually with or without our method.

J Raskin: How did you choose your widget set?
R. Jezek: We used windows 95 widgets.

J Raskin: Doesn't that guarantee a poor interface? [laughter]
R. Jezek: We had to start somewhere but other sets will be considered in future work.

J. Coutaz: You have focused on the mapping between domain objects and widgets. Another strategy could have been the mapping between tasks and widgets?
R. Jezek: We have an existing tool where scenarios are defined so we concentrated on scenarios with attached tasks.

M. Harning: Have you considered situation where a task like copy maps into multiple widget – a button, a drop-down menu a drag-and-drop operation.
a) do you consider a drag-and-drop action as a widget?
b) Did you consider situations where one task maps into multiple widgets?
R. Jezek: We focused on simple user interfaces, without drag-and-drop, but it is good input for future work.