A ZOOM-BASED SPECIFICATION TOOL FOR CORPORATE USER INTERFACE DEVELOPMENT

Thomas Memmel, Florian Geyer, Johannes Rinn, Harald Reiterer
Workgroup Human-Computer Interaction, University of Konstanz
Universitaetsstrasse 10, D-78457 Konstanz, Germany

ABSTRACT

We introduce a UI specification tool that interactively integrates interdisciplinary and informal modelling languages with different fidelities of UI prototyping. Our innovative experimental tool, called INSPECTOR, is the first tool that assembles models and design into a UI specification with a zoom-based visualization approach.

KEYWORDS

User Interface Specification, Interdisciplinary Modelling

1. INNOVATIVE USER INTERFACE SPECIFICATION APPROACH

In our research we analyze user interface (UI) specification processes in industrial practice [Memmel et al. 2007a]. Typically, UI specification teams are made up of actors from various disciplines. We identify a lack of UI tools that allow all actors to participate in a collaborative UI specification process. Many existing UI tools are focused on the formalisms required to automatically generate the UI. This prevents the current tools from adequately supporting the thought and design tasks that experts have to accomplish in order to create usable, effective and innovative UIs. Moreover, actors with no IT background are usually not educated in formal modelling (e.g. UML). Unfortunately, office applications therefore overwhelmingly dominate as tools for writing requirements (Word, Excel), drawing informal diagrams (Visio) and creating static UI mockups (PowerPoint). But their application is uncoordinated and not standardized, and the artefacts are usually simply uploaded to a requirements management tool such as Doors. Hence, requirements remain static, as mostly unrelated aggregations that cannot be evaluated until programmers begin to use completely different tools. In a black-box process they deliver coded parts of the system. But simulations of look and feel are necessary to trace how requirements will be translated into UI design and to identify missing or incomplete requirements. With our approach we bridge the disciplines by providing shared means of modelling requirements and UI design. In order to make the process of UI specification more transparent and traceable, we want to supplement UI prototypes with the requirement models they are based upon. When actors can browse from UI to underlying models, the relationships and interdependencies of requirements and design become more obvious. By integrating models and UI design, we create interactive UI specifications. Based on interviews and workshops, we selected models that represent a common denominator between human-computer interaction experts, software engineering specialists and business modellers. With regards to agile methods, we identified informal models for UI specification that can be easily understood and applied by all actors [Memmel et al. 2007b]. Among them are personas, role maps, task maps, use-case diagrams, essential use cases, and (data) flow and activity diagrams. We choose models that can be interrelated in terms of travelling from abstract descriptions to graphical notations. For prototyping the UI, we provide the chance to create designs with different grades of fidelity. We suggest scenarios as the entry point and we use a UI storyboard to enable access to the modelling and UI design layer (see Figure 1).

2. IMPLEMENTATION AND DEMONSTRATION

In this context, we developed an experimental tool to drive modern UI specification processes. INSPECTOR is the first tool that combines models and design into a UI specification with the help of a zoom-based

visualization approach. The UI specification can be interactively experienced and provides simulations of the UI behaviour. INSPECTOR is built in C# (over 10,000 lines of code) utilizing the .NET framework (Microsoft) and the Piccolo framework (http://www.cs.umd.edu/hcil/jazz/). Our system demonstration shows how actors use INSPECTOR to create, edit and relate all artefacts to assemble a UI specification. These activities include the demonstration of different zoom and pan interactions. For example, the actor can zoom from the UI storyboard into UI designs and create connections to build an executable simulation. Semantic zooming into objects on the canvas reveals more information and provides appropriate tools for modelling or design. Animated zooming will help to understand the topology of the overall specification space and carry the actor between models. An overview window functions as a navigation aid and helps to maintain orientation. An object navigator can be used to invoke jump zooms to quickly navigate to objects far away on the canvas in terms of the zoom and pan operations otherwise required to reach them. The demonstration will show how models are linked to UI designs, underlying or superordinate models, or both. This includes a demonstration of all the means of expression that our tool provides.

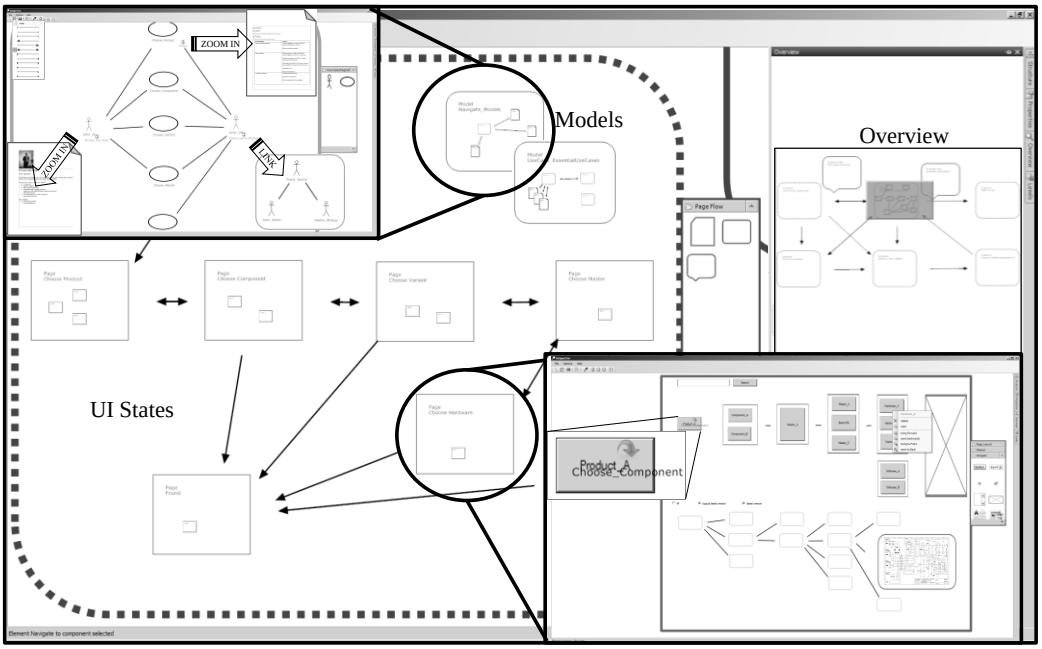

Figure 1: Screen from INSPECTOR, showing the UI storyboard with shapes for UI design and models (magnified)

3. CONCLUSION

The main contribution of INSPECTOR is the new approach of integrating models and UI design. It is the intermediate solution between computer aided sketching tools and formal modelling and programming tools. In first reviews, the ability to interactively browse UI specifications and the drill-down from UI design to underlying models, which adds traceability and provides the opportunity for quality assurance at any stage, were highly appreciated. We show that designing innovative systems starts with providing innovative tools.

REFERENCES

Memmel, T.; Bock, C.; Reiterer, H (2007a): Model-driven prototyping for corporate software specification, In Proceedings of the Engineering Interactive Systems Conference (EIS 2007), Salamanca, Spain (to be published)

Memmel, T.; Gundelsweiler, F; Reiterer, H. (2007b): Agile Human-Centered Software Engineering, In Proceedings of the 21st BCS-HCI, pp. 167-175