TOOL-SUPPORT FOR INTERDISCIPLINARY AND COLLABORATIVE USER INTERFACE SPECIFICATION

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

ABSTRACT

When the user interface (UI) has to be specified, a picture is worth a thousand words, and the worst thing one can do is attempt to write a natural language specification for it. Nevertheless, this practice is still common, and it is therefore a difficult task to move from text-based requirements and problem-space concepts to a final UI design, and then back again. Especially for the specification of interactive UIs, however, actors must frequently switch between high-level descriptions and detailed screens. In our research we found that advanced UI specifications therefore have to be made up of interconnected artefacts that have distinct levels of abstraction. With regards to the transparency and traceability of the rationale of the UI specification, transitions and dependencies must be visual and traversable. We introduce a model-based UI specification method that interactively integrates interdisciplinary and informal modelling languages with different fidelities of UI prototyping to an interactive design rationale. With an innovative experimental tool we assemble models and design to an interactive UI specification. With a zoomable user interface (ZUI) approach, we can visualize the modelled artefacts and the overall UI specification space on desktop computers as well as on megapixel displays.

KEYWORDS

User interface requirements modelling, participatory design, cooperative design, usability, user interface specification

1. INTRODUCTION

It is generally accepted by both software practitioners and human-computer interaction (HCI) specialists that structured approaches are required to model, specify and build interactive systems with high usability (Metzker and Reiterer 2002). Nevertheless, in many organizations UI design is still an accidental by-product and HCI methods are not sufficiently embedded in the overall process. A lack of communication among actors from different disciplines leads to misunderstandings and bad design decisions. When collaborative and concurrent development of design alternatives is hampered, a great potential for successful UI designs is not utilized. This phenomenon can be explained by the fact that, in particular, especially formal UI tools and requirements modelling languages prevent actors from taking part in designing interactive systems if they do not have adequate knowledge of specific terminologies. Moreover, many tools turn out to be more focused on requirements management than on providing support in extracting requirements from user and task needs and translating them into good UI design. After all, despite - or perhaps precisely because of - the vast functionality of many tools, the outcome is often unsatisfactory in terms of UI design, usability and aesthetics. This is described as the high threshold - low ceiling phenomenon (Campos and Nunes 2004).

1.1 Shortcomings and change of current UI specification practice

Over the last 3 years, we observed UI development practice in the German automotive industry (Memmel et al. 2007; Memmel and Reiterer 2008). As a consequence of the lack of appropriate tools, many actors tend to use tools they are familiar with, which has been well observed by Campos and Nunes (2004). In addition, we identified different populations of tool users, which can be assigned to two different main areas of corporate UI development projects:

1. Client. Business personnel, marketers, domain experts and HCI experts use Office-like applications such as Word, PowerPoint or Visio to document user needs and context of use in order to define the problem-

space. They will then translate the needs as analyzed, and their contextual conditions, into general usage requirements - like task and user models - and evaluate their work at several quality gates. At this stage, responsibility is typically shared with, or completely passed on to, an IT supplier.
2. Supplier. Actors with a sophisticated IT and software engineering (SE) background (e.g. programmers or designers) translate usage requirements into UI and system requirements, deliver prototypes and conclude the outcome in a UI specification. Working with UI builders, and using more formal, precise and standardized notations, they narrow the solution space towards the final UI.

Collaboration and communication is essential to promote UI design decision. But the difference between these groups of actors and their favourite tools tends to result in a mixture of formats. This makes it difficult to collaboratively promote concepts and creative thinking without media disruptions and loss of precision (Memmel et al. 2007). The following negative factors therefore contribute to UI specification and development failure:
1. The lack of a common course of action and the use of inappropriate, incompatible terminologies and modelling languages (Zave and Jackson 1997) that prevent even the minimum levels of transparency, traceability and requirements-visualization necessary for successful collaborative design.
2. The difficulty in switching between abstract and detailed requirement models as well as UI designs due to a lack of interconnectivity (compare Campos and Nunes 2006).
3. The difficulty of travelling from problem space (requirements) to solution space (UI design), a difficulty that turns the overall UI development into a black-box process.
4. The lack of support in visualization and sharing of complex networks of requirements and modelling artefacts in creative environments (Beyer and Holtzblatt 1997). Insufficient (i.e. static, non-interactive) means of communication and cooperation lead to misunderstandings and wrong design decisions.
5. The burial of critical requirement information in documents that are difficult to browse and have very awkward traceability. The resulting misconceptions lead to costly change requests and iterations, which torpedo budgets and timeframes, and endanger project goals.
6. The lack of feedback support in UI tools and in requirements-modelling practice. To achieve a comprehensive design rationale, all decisions and feedback on design alternatives have to be recorded for future reference.

Furthermore, in our research for Porsche AG and Daimler AG we found the following sticking points that tend to change common UI specification processes in large organizations (Memmel et al. 2007):
1. Due to the strategic impact of many software products, clients want to increase their UI-related competency in order to reflect corporate values by high UI quality.
2. Whereas conceptual modelling, prototyping or evaluation have always been undertaken by suppliers, the client himself now wants to work in the solution space and needs to develop the specification in-house.
3. Actors want to establish a common understanding about the UI they have to specify. Design decisions and tradeoffs, as well as the arguments that led to those decisions, must be recorded in order to make the process transparent and traceable. Promoting creativity becomes a main objective during the collaborative specification of UIs with high usability.
4. For the client to gain flexibility in choosing his suppliers, the role of the supplier becomes limited to programming the final system. The client gains a timetable advantage from this change and, by having an in-house competency in UI-related topics, the client becomes more independent and can avoid costly and time-consuming iterations with external suppliers.

1.2 Requirements for adequate tool support

The shortcomings and drawbacks of the previously described work practice have to be addressed by an integrating tool that allows the translation of needs into requirements and subsequently into good UI design. All artefacts must be incorporated in a design rationale, out of which the interactive specification is built. It is also essential to allow actors to discuss conceptual designs, to make decisions and to consider tradeoffs. Therefore, an adequate tool must also provide means to support collaborative work during creative meetings and review sessions. Table 1 shows an overview of relevant tool requirements that gave direction to our approach. In this paper we present both a set of models and a corresponding tool named INSPECTOR, still

under development, which are designed to support interdisciplinary teams in gathering user needs and task information, translating these into UI-related requirements, designing prototypes of different fidelities and compiling the resulting artefacts to an interactive UI specification. The term interactive refers to the concept of making the process visually externalized to the greatest extent possible. Being interactively connected, all of the ingredients result in a compilation of information items that are necessary to specify the UI. In Section 2 our research is linked to related work. Section 3 initially presents the common denominator in modelling that we developed. There then follows a detailed explanation of how INSPECTOR will utilize the resulting interconnected hierarchy of notations to allow interdisciplinary, collaborative work. We then illustrate how UI designs can be created and exported in machine-readable formats such as XAML (www.microsoft.com). Section 4 shows how INSPECTOR is applied as a collaborative specification tool on a megapixel display. The results of two evaluation studies are presented in Section 5. The paper ends in Section 6 with a summary and an outlook.

Table 1: Requirements for UI tools for interactive UI specification (outline from Memmel and Reiterer 2008)

Tool Requirement	**Added Value**
Concentration on a specific (agile) subset of modelling artefacts allowing actors from different disciplines to use familiar notations and models that best leverage collaboration.	Provide support for design assistance and creative thinking for everybody; all actors can proactively take part in the UI specification process.
Visualization and interaction concepts that keep artefacts accessible and support interdisciplinary cooperation; representation of common problem and solution space.	Bridge differences in work practice among disciplines; Eliminate diversity of formats, promote traceability and accessibility for meetings.
Sharing of workspaces and feedback between actors and teams; support actors during creative sessions.	Facilitate communication and decision making; provide overview over design process and status; allow common perception of design rationale.

2. RELATED WORK

Campos and Nunes (2004) presented the tools CanonSketch and TaskSketch. CanonSketch supports abstract UI prototyping and is based on the concept of canonical abstract prototyping. In contrast to CanonSketch, we also support detailed UI prototyping because we found that the high-fidelity externalization of design vision is especially important in corporate UI design processes. TaskSketch also supports collaborative modelling, but focuses on linking and tracing use cases, by means of which it significantly facilitates development tasks with an essential use-case notation. It provides three synchronized views: the participatory view uses a post-it notation to support communication with end-user and clients; the task-case view is targeted towards designers and is a digital version of index cards; the UML activity diagram view is adequate for software engineers. Both tools therefore share some basic concepts with our approach. Using our proposed set of models, however, the actors are able to choose from a wider range of means of expression, both on a more detailed level and on some levels that are more abstract than essential use-cases.
According to the CAMELEON reference framework (Calvary et al. 2003), we considered different layers for tasks and concepts, abstract UI design, detailed UI design, and the final UI. For the utilization of an electronic whiteboard metaphor as well as for visualization concepts for our UI specification space, we were inspired by the tool DAMASK (Lin and Landay 2002). It uses a ZUI approach for switching between different levels of UI detail through a visual drill-down process. A ZUI-based whiteboard also fits very well with the idea of electronic whiteboard collaboration as suggested by Dix et al. (2003).

3. INSPECTOR: INTERDISCIPLINARY SPECIFICATION TOOL

Modelling UI requirements and creating UI specifications is a highly interdisciplinary process in the industry. Actors from different disciplines have to contribute their ideas and need to externalize their vision. But it is difficult for them to recognize the meaning of unfamiliar models outside the scope of their discipline. Without a shared understanding, creative thinking is limited and promoting design decisions is difficult. For our specification tool, called INSPECTOR, we followed a 3-step approach to define an adequate tool.

Firstly, we analyzed the characteristics and disciplines of all actors in the supply chain (see Section 1) to identify opportunities for better cooperation. With a discipline-specific point of view, the project population can be classified into three main groups. Specialists in business process modelling (BPM) are engaged for the analysis, conception, and development of information systems as well as the corresponding business-process reengineering (Malhotra 1998). HCI experts are employed for dealing with the UI design, while SE professionals are concerned with background events and changes of states within the system. On the one hand, the currently dominating Office-like artefacts are easy to understand, but ambiguous and insufficient for expressing behavioural aspects of an interactive UI. On the other hand, the Unified Modelling Language (UML) is also inappropriate, because its formal patterns are not understood by all actors. We therefore developed a common denominator of UI requirement models (Memmel and Reiterer 2008). With a thoughtfully selected and reduced set of hierarchically classified models, we can ease the application of adequate UI modelling languages (see Section 3.1ff).

Secondly, we searched for adequate means to visualize the corresponding models and allow collaborative access to all items in the UI specification. Keeping created artefacts visible to all actors enhances creativity, supports communication, makes it easier to achieve a common design vision and leads to faster decision making (Beyer and Holtzblatt 1997). Our observation of work practice in the automotive industry revealed that whiteboards are familiar to all actors and enhance creativity by offering informal sketching methods. We therefore decided that an innovative UI specification tool could best address the demand for a creative environment by incorporating an electronic whiteboard metaphor. An electronic whiteboard allows the creation of rich content such as text, pictures, diagrams or even videos and can be visualized at different scales (see Section 4). Moreover, created artefacts can be saved easily and contribute to the safekeeping of the design rationale. Because all design artefacts must be permanently accessible to enhance traceability, promote transparency, ease view transformations and allow collaboration (Beyer and Holtzblatt 1997), all artefacts are consequently arranged on one single canvas. Visual links and references between them enable a roundtrip-engineering between problem- and solution-space.

Thirdly, we decided to employ a ZUI to implement the whiteboard metaphor for our experimental tool. The overall interaction can be compared with moving in different directions in front of a large whiteboard, plus stepping back for overview or getting closer for the details. We wanted to provide users with a feeling of *diving* into the information space of the UI specification artefacts and therefore composed a notation of graphical objects that represents different levels of abstraction within our modelling hierarchy (Memmel and Reiterer 2008). The latter can be presented and navigated top-down or bottom-up, which is very well supported by ZUIs (Pook et al. 2000). Altogether, the appearance of INSPECTOR's whiteboard is based on a linear scaling of objects (geometric zooming) and on displaying information in a way that is dependent on the scale of the objects (semantic zooming) (Ware 2004). Automatic zooming organizes selected objects on the whiteboard to support the user in exploring the topology of the information space and in understanding relationships. For switching between abstractions within and between models and UI designs, the user can manually zoom in and out or pan the canvas. During modelling, or while traversing relationships by panning and zooming, hints about the current zoom factor and the current position in the information space are given in order to avoid disorientation. A common way of supporting the user's cognitive (i.e. spatial) map of the information space is an overview window (see Fig. 2). In addition, INSPECTOR provides a tree structure control as navigation support, which allows zooming into areas far removed from the current focus and helps to understand the hierarchy of artefacts. The creation and manipulation of artefacts on the whiteboard is facilitated by direct manipulation methods such as drag & drop. This interaction style is suitable for creating all required models, as it is easy to use, straightforward and consistent throughout all levels of interaction. To create specification artefacts, the user simply drags & drops graphical objects (i.e. shapes or UI widgets) from a context-sensitive toolbox onto the whiteboard. By zooming into created shapes with a double-click (automated zoom) or by complementary scrolling and panning operations (smooth zooming) with the mouse-wheel, the user drills down into the hierarchically arranged specification space. Every artefact will provide its means for manipulation once a certain zoom-level has been reached (semantic zoom). The ZUI provides infinite screen space for an extensive number of UI specification artefacts, is perfectly suited for face-to-face collaboration in meeting rooms and unleashes its real potential on large screens with high resolutions (see Section 5). In this way, multiple actors can view and discuss the design space and its artefacts and relations at the same time. Zooming into artefacts allows focusing for discussion and feedback.

In the following, we present the different layers of our ZUI-based specification tool by going through a level-wise explanation of a Daimler case study.

3.1 Scenario map: text-based notations of needs and requirements

As an interdisciplinary modelling language, research suggests the use of scenarios (Barbosa and de Paula 2003). On starting INSPECTOR, the user first creates a scenario map that shows all scenarios that will be modelled (see Fig. 1, left). On the one hand, the map consists of information bubbles that describe the problem scenario (see Fig. 1, right). Therefore, INSPECTOR provides a build-in text editor with appropriate templates and also enables the direct integration of existing requirement documents into its repository. On the other hand, the user can zoom-in to the scenario containers in order to access the storyboard layer, where the scenario is worked out with graphical requirement notations and UI designs.

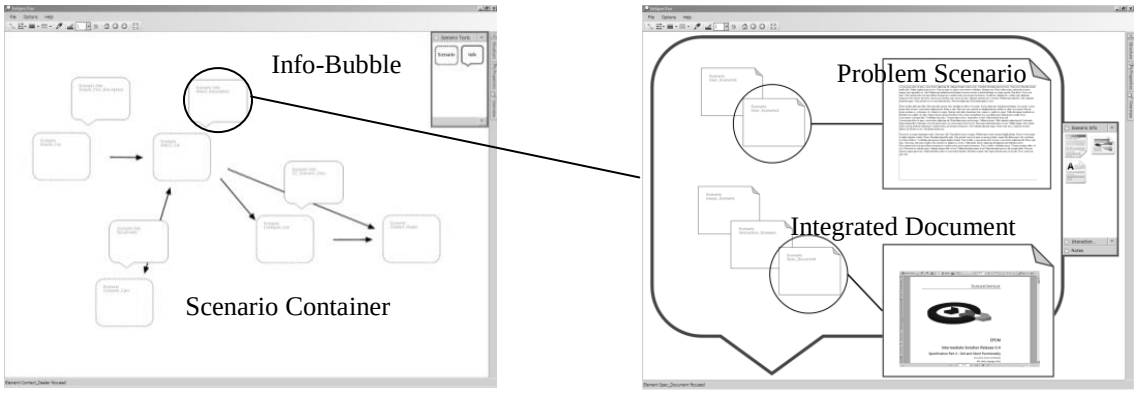

Fig. 1: Scenario map as entry stage to the modelling process (left); scenario info-bubble (right)

3.2 Storyboard layer: associating models with UI design

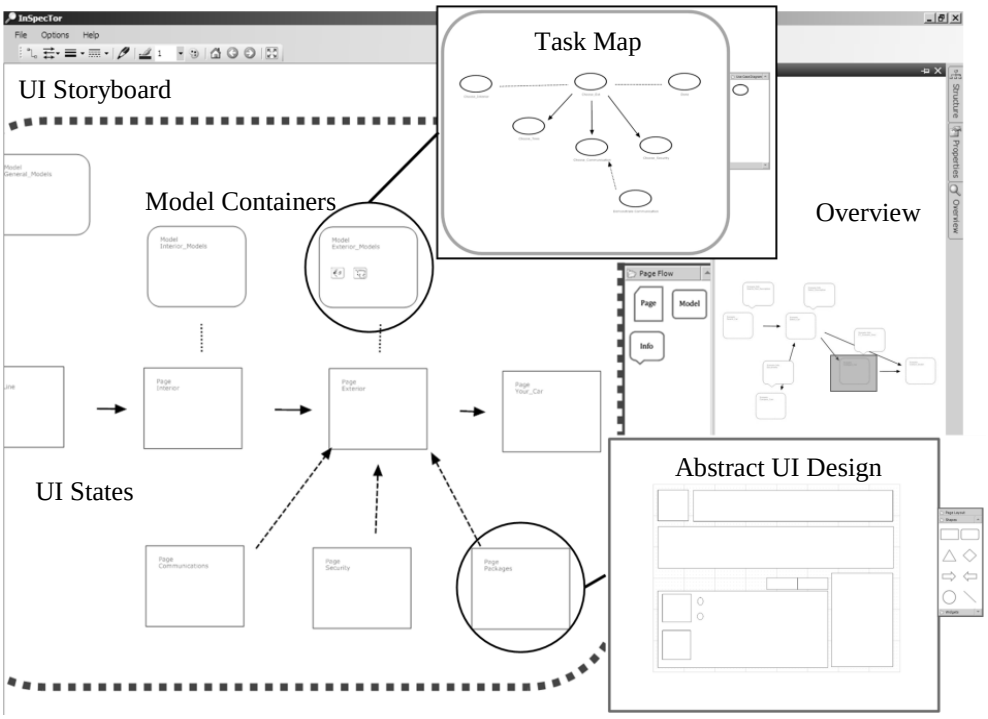

Fig. 2: Storyboard layer with UI design and models; magnified area shows abstract UI design prototype (bottom right) and task (use-case) map (top right). Also shown is the overview panel (right)

For the storyboard layer we decided to keep the typical UI storyboard notation we know from HCI (Beyer and Holtzblatt 1997). At this zoom-level, the user is offered artefacts that support the process of translating needs into requirements and also into UI design. The storyboard serves as interface layer between needs, requirements (models) and the associated UI design (pages). Actors create and connect containers, which group a set of models or page designs (i.e. UI states). By arranging and connecting several page containers on the whiteboard, a UI storyboard that represents the dialogue flow is created (see Fig. 2) Actors can describe the requirements related to pages by zooming into the model containers (see Section 3.3), whereas zooming into page containers enables modelling of the different UI states at the design-layer (Section 3.4).

3.3 Modelling layer: user, task, and interaction modelling

The modelling-layer offers containers to describe users, tasks and interactions with textual and graphical notations. For describing users and their needs, HCI recognizes user profiles, (user) scenarios (Rosson and Carroll 2002), role models (Constantine and Lockwood 1999), and personas (Beyer and Holtzblatt 1997). Roles and personas are also known in SE and BPM and are therefore appropriate for initial user-needs modelling. Role maps (Constantine and Lockwood 1999) help to relate user roles to each other, and personas provide means to understand future users. Although different in name, task cases (HCI), essential-use cases (SE), and business-use cases (BPM) can all be expressed in a classical use-case notation that models tasks and requirements (see Fig. 3, left). Moreover, use-case diagrams (SE, BE) overlap with use-case and task maps (HCI). Linking the actor shape to personas enables a zoom-in from the use-case diagram to the corresponding persona. Similarly, each case shape can be linked to e.g. an essential use-case description. Then, the zoom-in on the case shape navigates the user to a detailed description of user intention and system responsibility according to the use case. We also included activity diagrams (see Fig. 3, right) which are typically used for business-process modelling, for modelling the logic captured by a single use-case or usage scenario, or for modelling the detailed logic of a business rule. They are the object-oriented equivalent of flow charts and data-flow diagrams. Data-flow diagrams model the flow of data through the interactive system (Memmel and Reiterer 2008). They are more formal than the models HCI experts are usually familiar with, but they therefore extend the expert's competency in interdisciplinary modelling.

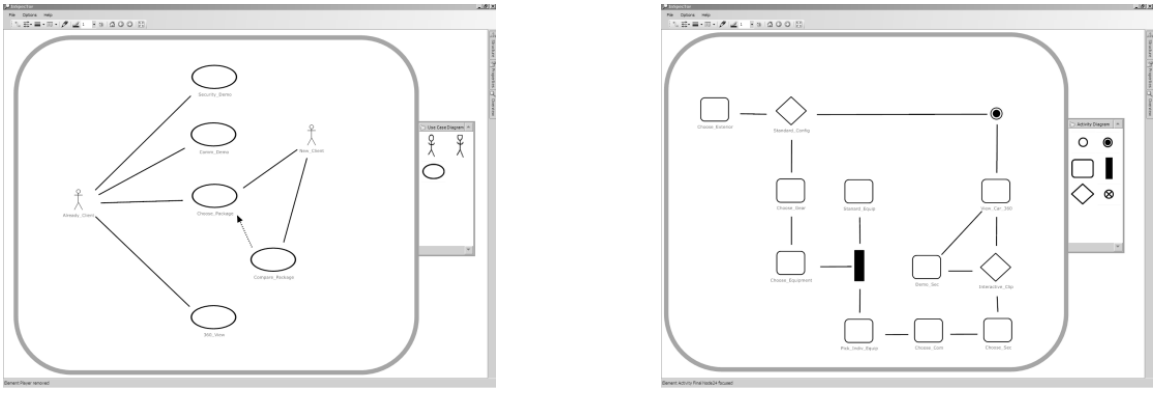

Fig. 3: Use-Case Diagram (left); Activity Diagram (right)

3.4 Design layer: abstract prototyping and detailed look and feel specification

By drilling down to the design layer the user is able to create UI designs within INSPECTOR, and these will be used for iterative or alternative UI development. This allows the designing and evaluation of the UI at early stages while supporting traceability between requirements to design. The design layer within INSPECTOR provides a wide range of prototyping tools from low-fidelity methods like sketching, drawing and simple shapes, to medium-fidelity methods, like wireframes or placeholders and canonical components (Constantine and Lockwood 1999), to high-fidelity widgets like common UI controls (Buttons, Checkboxes, etc.), images and embedded objects such as Macromedia Flash (see Fig. 4, left).

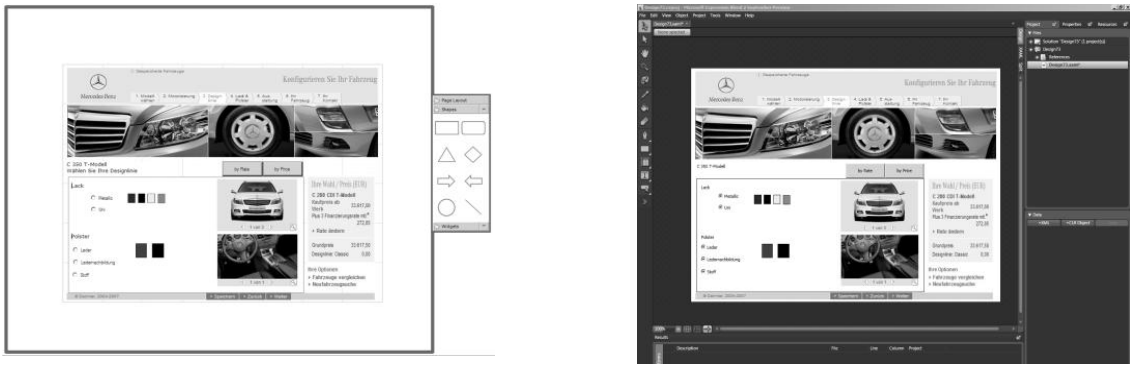

Fig. 4: INSPECTOR-made hi-fi UI design (left) in Microsoft Expression Blend (right)

UI elements are then locally linked to other UI states within the storyboard layer (see Fig. 2) to allow navigation within a prototypical simulation, which serves as the executable, interactive part of UI specification and makes the package complete. From here on, the actor can later explore, create and change models by zooming back to the relevant area of the UI specification. Moreover, programmers can pop-up the interactive UI prototype to get guidance on the required UI properties. An interactive simulation can be created once all UI states on the storyboard are properly linked and contain at least one UI design version. Being exported into the XAML format, it is possible to execute a prototype of the UI in a web browser such as Microsoft Internet Explorer. The XAML export also allows the specified UIs to be reused with GUI builders such as Microsoft Expression Blend (see Fig. 4, right).

4. COLLABORATIVE UI SPECIFICATION AT THE POWERWALL

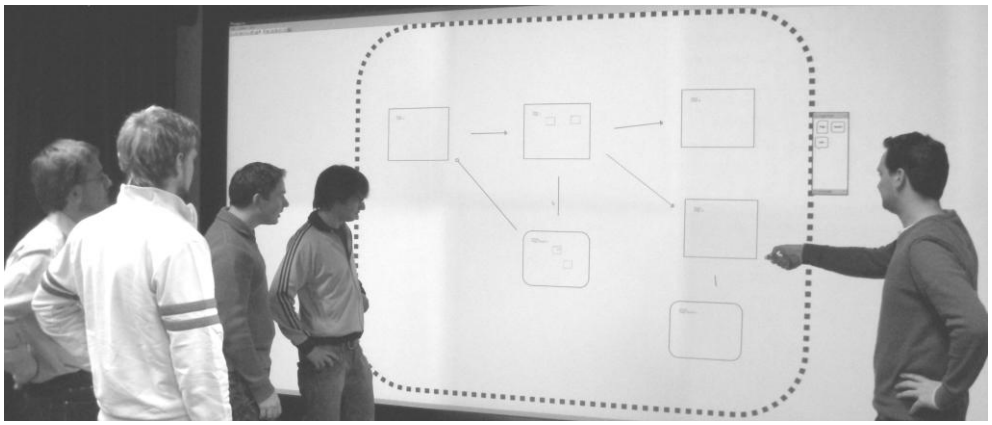

Fig. 5: Utilizing INSPECTOR for collaborative meetings at a megapixel *powerwall* with laser-pointer interaction

In meeting or decision room set-ups, INSPECTOR supports collaboration and decision-making. Users can cooperatively work on requirement models or UI designs during brainstorming sessions. Utilized as an electronic whiteboard, INSPECTOR records all created artefacts in a structured manner. Actors can also work asynchronously using their own workspace e.g. on a desktop installation. Modelled artefacts are then exported into XML documents and re-imported into a shared workspace, which resembles the common design rationale. First experimental setups with our high-resolution powerwall installation (4640 x 1920 pixels) allowed a comprehensive view on our zoomable specification space (see Fig. 5). This high-resolution display supports our ZUI approach by displaying a wide range of artefacts and relations (overview) to all actors. Laser-pointer interaction, which was also developed in our work group and presented at the last IHCI conference (König et al. 2007), enables easy-to-use cooperative interaction style. Through point and click operations, actors explore and manipulate the UI specification space.

5. EXPERT FEEDBACK AND USABILITY STUDY

We have started to interview software and UI specification experts (n=12) from Daimler in a questionnaire-based usability study. The participants were introduced to INSPECTOR through a short demonstration, a video and a supplementary text explaining the motivation for our approach. Each expert was provided with an installation of the tool and had two weeks to return his feedback by means of a questionnaire that was divided into 5 parts. The first part was designed to (1) identify the field of activities of every respondent, (2) get an overview of the models and tools typically applied, and (3) get an assessment of difficulties along the supply chain. The second to fourth parts asked about INSPECTOR in terms of (1) the applicability of the modelling notations, (2) the completeness of the UI design capabilities and their practicability for UI evaluation, and (3) the assessment of the tool's general usability and the user experience provided. The fifth part asked if INSPECTOR could, in general, improve the UI specification practice.
Currently, half of the questionnaires have been completed (n=6) and we can provide a first outline of the most important results (see Table 2). So far, all respondents have stated that INSPECTOR, as a tool that combines models with UI Design, contributes great value to their work style (average 4.83 pts; on a 5-point Likert scale). The added value was particularly identified in terms of an increased coherence of models and design artefacts, whereby INSPECTOR enhances traceability and transparency. The very early version of INSPECTOR was therefore already expected to be able to improve existing UI specification practice (average 3.83 pts). The participants of the study where quite satisfied with INSPECTOR`s support for text-based and graphical requirements modelling (average 4.00 pts). Nevertheless, the feedback pointed out to the necessity for a better linking functionality between the modelling artefacts. Consequently, we implemented a visualization that highlights all outgoing and incoming links of a model in order to enhance traceability.

Table 2: Overview on feedback from Daimler experts (outline); average points based on a 5-point Likert scale

Questionnaire topic	**Avg.**
Ability to integrate documents and logic with INSPECTOR	3.66
Chance to capture conceptual and schematic ideas	3.83
Support for user, task and interaction modelling	4.00
Possibility to link models and to thereby increase the traceability and transparency	3.66
Text-based and graphical requirements modelling (aggregated)	**3.79**
Accessibility of the prototyping features	**3.16**
Provided functionality at the UI design layer	**3.40**
Applicability of the UI designs for usability evaluations	**3.33**
Possibility to link UI designs in order to create a simulation	3.25
Overall UI prototyping capabilities (aggregated)	**3.28**
Chance to get both overview and detail on the zoom-based specification space	3.33
Helpfulness of the zoom-interaction style during prototyping and modelling	3.00
Support for switching between created artefacts	3.50
Accessibility of all necessary information on the zoom-canvas	3.50
Overall rating of the interaction with INSPECTOR (aggregated)	**3.33**
The overall contribution of INSPECTOR to existing UI specification practice	**3.83**
The improvement of work style through a combination of different models with multi-fidelity UI design	**4.83**

Altogether, as we had expected from our first evaluation study, the results also highlighted chances for improvement. Due to the experimental stage of INSPECTOR's design and prototyping facilities, the experts missed some important features such as master components and templates. These are needed to allow for rapid prototyping and quick generic changes. Besides a required copy & paste mechanism for the UI design layer, we therefore implemented support for grouping UI elements and storing them in a template repository. In order to improve the utility of INSPECTOR during usability evaluations of modelling and design artefacts, we developed an annotation component (see Fig. 6). During meetings, discussions and feedback sessions, sticky notes can be attached to all artefacts on the whiteboard. This allows the recording of feedback and design decisions for later consideration during subsequent specification tasks. The notes can be accessed in a spreadsheet component which allows sorting and filtering, as well as jump navigation towards them.

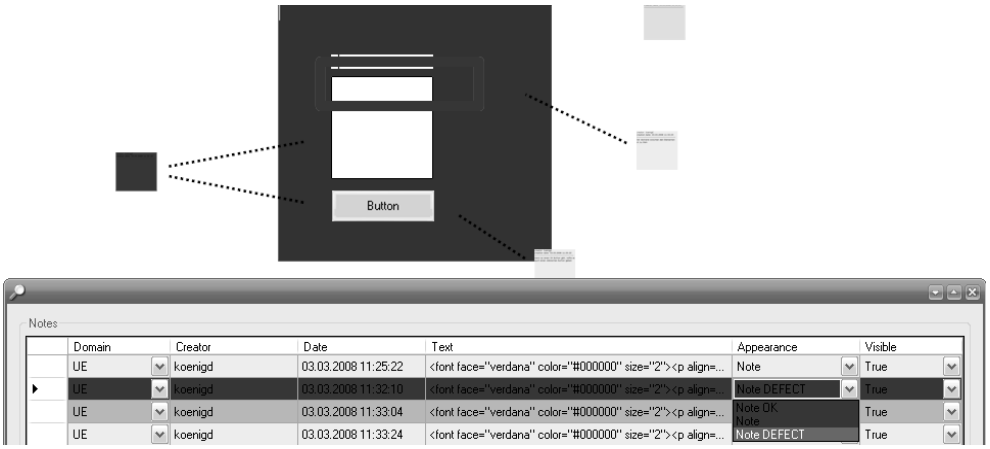

Fig 5: Annotations for feedback (top) are organized in a management console (bottom).

Furthermore, some experts stated that during creating a UI design, the interaction with INSPECTOR could be enhanced by a contextual layer (Pook et al. 2000). This could give the expert the chance to easily cross-check the design with underlying models. Instead of frequently jumping back and forth on the ZUI canvas, it should be possible to temporarily visualize models and UI concurrently. In order to improve the usability of INSPECTOR, we have therefore started to develop a mechanism that allows previewing the requirement model(s) linked to a UI design and vice versa.

Other usability issues concerned the general interaction with the tool and were similar to those found during a diary study. For the latter, we used INSPECTOR during an interaction design lecture. Three groups of computer science and HCI students (n=8) were asked to use the tool during a use-case study on the specification of rear-seat entertainment systems. For a period of three weeks, every student wrote his own diary to give insight into (1) the kind of models created, (2) additional tools that were applied, (3) problems that occurred, (4) ratings of the user experience, (5) general issues and opinions about the tool. We decided on the diary study in order to evaluate INSPECTOR over a longer period of time. Because we were interested in how the empirical results change with the duration and intensity of usage, we preferred a long-term study to classical usability tests. In weekly workshops, we discussed the intermediary results and recorded the issues for subsequent correction. By means of the diary study, we found, for example, that objects on the ZUI canvas occasionally behaved inconsistently after the tool was used for several hours and an extended amount of zoom operations had been performed. Students also reported issues with integrated external documents (PDF, Word, etc.), when these were repeatedly saved and opened. This led to a disarrangement of the XML structure in saved project files and significantly prevented a fluent and enduring work style. To have identified these problems in a much shorter lab-based usability study would have been pure chance. Thanks to the diary study, we were able to solve these issues quickly. Moreover, we found that some participants preferred to create the first abstract prototypes initially with paper and pencil. We realized that the use of the built-in sketching mechanism increased as soon as we provided a pen tablet as an input device. In addition, it proved to be very difficult to rapidly prototype UIs with point and click interaction on the canvas. We will therefore evaluate different pen tablet technologies that we could constantly combine with INSPECTOR. This will significantly increase the application performance during design sessions. In addition, students were initially not comfortable with all the notations provided and required assistance on their proper application. We addressed this issue by making a start on including a help feature that explains notations as well as their scope of application. In addition, we enhanced the affordance of templates for personas or essential-use cases, for example, to ease the understanding of the artefacts.
Ultimately, the diary study and the upgrades resulted in an improvement of the feedback on the tool usability: rated with an average of 1.75pts (std. 0.46) (on a 5-point Likert scale) after the first week and 3pts (std. 0.00) after the second, participants re-viewed INSPECTOR with an average of 4.25pts (std. 0.46) at the end of the study. A repeated-measure ANOVA revealed a significant main effect for the rating across the weeks ($F(2,14)=105.00$, $p<0.001$). Furthermore the differences between each week are also very significant statistically (week 1 vs. week 2: $F(1,7)=58.33$, $p<0.001$; week 2 vs. week 3: $F(1,7)=58.33$, $p<0.001$).

6. SUMMARY AND CONCLUSION

Based on our experience in UI specification and design, we have come to the conclusion that the typical methods and tools available are not adequate. UI tools must support not only the "hard" aspects, but also the "soft" aspects of UI development to support the delivery of usable systems in the future (Campos and Nunes 2006). These include support for collaboration, creativity and improvisation. We focused our research on actors in charge of the conceptualization, and particularly the specification, of innovative UIs with high usability. With our experimental tool-design, actors are supported in applying informal models, and are given the opportunity of UI prototyping with different fidelities. Being logically linked, transitions from abstract to detailed artefacts increase the transparency of design decisions and enhance the traceability of dependencies. This improves communication, consistency, and lastly, the necessary understanding of the overall problem space. Based on a ZUI approach, our INSPECTOR tool integrates and innovatively interconnects the required artefacts in a visual UI design rationale that can be interactively experienced. First evaluation studies demonstrated the fruitful contribution of our approach and we already plan additional interviews with experts. We will therefore continue to enhance our tool in order to upgrade HCI processes and to make it an innovative and fully capable alternative for the tool-landscape that we found in current industrial practice.

REFERENCES

Barbosa S. and de Paula M., 2003. Interaction Modeling as a Binding Thread in the Software Development Process. *In ICSE Workshop on SE-HCI*, Portland, USA. pp. 84-91.

Beyer H. and Holtzblatt K., 1997. *Contextual Design: A Customer-Centered Approach to Systems Designs (Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann Publishers, San Francisco, USA.

Calvary G. et al., 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* Vol. 15, No. 3, pp. 289-308.

Campos P. and Nunes N. J., 2004. CanonSketch: a User-Centered Tool for Canonical Abstract Prototyping. *Proceedings of the EHCI-DSV-IS*. Hamburg, Germany, pp. 146-163.

Campos P. and Nunes N. J., 2006. Principles and Practice of Work Style Modeling: Sketching Design Tools, *Proceedings of HWID'06 - Human-Work Interaction Design*, Madeira, Portugal, pp. 203-219.

Constantine L. and Lockwood L., 1999. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design (ACM Press)*. Addison-Wesley Professional, Boston, USA.

Dix A. et al., 2003. *Human-Computer Interaction (3rd Edition)*. Prentice Hall, New York, USA.

König W. A. et al., 2007. Position-independent interaction for large high-resolution displays. *Proceedings of IADIS International Conference on Interfaces and Human Computer Interaction*. Lisbon, Portugal, pp. 117-125.

Lin J. and Landay J. A., 2002. Damask: A Tool for Early-Stage Design and Prototyping of Multi-Device User Interfaces. *Proceedings of the 8th Int. Conference on Distributed Multimedia Systems*. San Francisco, USA, pp. 573-580.

Malhotra, Y., 1998. Business Process Redesign: An Overview, *IEEE Engineering Management Review*, Vol. 26, No. 3, pp 27-31.

Memmel T. and Reiterer H., 2008. Inspector: Method and tool for visual UI specification. *Proceedings of the 3rd IASTED International Conference on Human Computer Interaction (IASTED-HCI '08)*. Innsbruck, Austria, (to be published).

Memmel T. et al., 2007. Model-driven prototyping for corporate software specification. *Proceedings of the EHCI-HCSE-DSV-IS*.Salamanca, Spain.

Metzker E. and Reiterer H., 2002. Evidence-Based Usability Engineering. *Proceedings of the 3rd Conference on Computer-Aided Design of User Interfaces*, Valenciennes, France, pp. 323-336.

Pook S. et al., 2000. Context and interaction in zoomable user interfaces. *Proceedings of the working conference on Advanced Visual Interfaces*, New York, USA, pp. 227-231.

Rosson M. B. and Carroll J. M., 2002. *Usability Engineering: Scenario-based Development of Human-Computer Interaction*. Morgan Kaufmann Publishers, San Francisco, USA.

Ware C., 2004. *Information Visualization, Second Edition: Perception for Design. (Morgan Kaufmann Series in Interactive Technologies)*. Morgan Kaufmann Publishers, San Francisco, USA.

Zave P. and Jackson M., 1997. Four dark corners of requirements engineering. *ACM Trans. Softw. Eng. Methodol*, Vol. 6, No. 1, pp 1-30.