First publ. in: Proceedings of the Third IASTED International Conference on Human-Computer Interaction, March 17 - 19, 2008, Innsbruck, Austria / ed.: D. Cunliffe. Anaheim : Acta Press, 2008, pp. 180-185

USER INTERFACE SPECIFICATION IN COMPLEX WEB-BASED INFORMATION SPACES

Thomas Memmel[1], Heiko Ziegler[2], Richard Oed[2], Harald Reiterer[1]
[1]*Human-Computer Interaction Lab, University of Konstanz. Germany*
{memmel, reiterer}@inf.uni-konstanz.de
[2]*Daimler AG, Group Research & Advanced Engineering GR/ESP, Ulm, Germany*
{heiko.ziegler, richard.oed}daimler.com

ABSTRACT
The development of complex web-based information spaces faces very demanding challenges. Requirements analysis and design demands interdisciplinary modelling approaches. When using typical paper-based methods, an IT organization often experiences frustrating communication issues between teams as well as with suppliers. We present a toolkit for XML-based prototyping of corporate websites at Daimler AG. The approach beats a new path towards interactive specifications of user interface (UI) requirements and the substitution of paper-based artifacts.

KEY WORDS
User Interface Development, User Interface Software Tools, Architectures, and Technologies.

1. Introduction

From the authors' experience with the automotive industry in general and with Daimler AG in particular, we see that UI design departments become more important in interactive software systems specification. The risk of bad UI usability is considerable, and it is an economic risk. But integrating usability engineering (UE) during early stages of requirements engineering (RE) often causes conflicts among stakeholders and faces shrinking IT budgets and pressure of time. As discussed in [1], several ingredients can therefore contribute to software development failure: the increasing importance of the UI, the separation of professions, particularly software engineering (SE) and UE, and consequently a lack of methods, tools and process models that integrate all stakeholders. Consequently, new approaches to RE and specification practice are necessary [2]. In this article we introduce an UI prototyping and specification approach for the development of interactive corporate websites at Daimler AG. By employing prototypes as vehicles for UI specification, we are able to bridge communication and competency gaps. Furthermore, this makes the overall design process more efficient and effective, resulting in lower development costs, but improved quality. In Section 2 we summarize the importance of UE and point out the various challenges web usability and UI engineering processes have to face. We contrast the demands with the current shortcomings of wide-spread RE practice and propose a change of applied practice. Consequently, in Section 3 we present our toolkit for developing prototyping-based UI specifications of interactive websites. We illustrate a typical usage scenario in Section 4 and discuss the experiences with our toolkit in Section 5.

2. The corporate UI specification process

The UI is the part of the software that can help users to work more efficiently and effectively. When users are unable to perform their tasks, the usage of the software may be entirely incorrect, slow to make progress, and may finally lead to reduced acceptance and exploding costs. With corporate software, the UI also transports important (emotional) values such as corporate design (CD) and corporate identity (CI). At Daimler AG, a wide range of different software systems exists. In this article, we concentrate on web applications and exemplify our idea by the Mercedes-Benz digital sales channel. Websites play an important role in the market success of an automotive brand. The Mercedes-Benz website must create brand awareness, transport product values, enable product search and configuration, offer contact to retailers and is expected to increase customer loyalty. For each phase of the online customer lifecycle, such an information space has to provide appropriate pages and usable applications (e.g. car configuration, car search, car comparison).

2.1 Usual UI specification practice

Usually, the responsibility of Daimler AG as a client during software projects is narrowed to common RE practice: Different departments hesitate to agree upon functional requirements and ask potential end-users for their needs.

Konstanzer Online-Publikations-System (KOPS)
URN: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-75260
URL: http://kops.ub.uni-konstanz.de/volltexte/2009/7526

Afterwards, the functional concept is translated into a specification sheet. Consequently, at this stage functional and non-functional needs are extracted from narrative business requirements. This procedure also serves as quality gate and is very time-consuming. Various kinds of stakeholders are involved during these phases, e.g. account managers, sales personnel, marketing teams, ergonomists, technicians, suppliers, etc. A majority of stakeholders use different office applications for the specification of software systems [2]. This inevitably leads to a wide variety of formats that often cannot be interchanged without loss of precision or editability. Later on, domain experts hand-craft the UI specification. This operation is usually already supported by external agencies (e.g. consulting companies). Finally, the specification is forwarded to the supplier for implementation. Facing demanding timeframes and critical budgets on the one hand and the importance of its corporate websites on the other hand, typical web engineering processes at Daimler AG have to change. Considering the value of CD and CI, Daimler AG cannot restrict its responsibilities to the definition of user needs and functional requirements. Using abstract paper-based UI specification documents is a promising departure for every software project. But textual specifications usually fail to map real UI behaviour. They can only sketch out the *look* of a UI, but are unable to externalize the *feel*. Consequently, the assessment of user performance and user experience during actual interaction is postponed to later stages of design. This is too late, if the UI behaves inappropriately. Furthermore, written language is ambiguous and the lack of visual cues leaves room for misinterpretation. In addition, virtual prototypes cannot automatically be created from textual documents. Those who are responsible for actually coding the software system will use completely different tools during the implementation process. Consequently, the effort invested in drawing PowerPoint slides or Excel sheets, helps neither prototyping, nor the implementation of the final system. When interactive behavior has to be specified, a picture is worth a thousand words and "[…], the worst thing that any project can do is attempt to write a natural language specification for a (UI)" [3].

2.2 From UI prototyping to UI specification

Setting up own know-how at Daimler AG in important areas such as interaction design (UE), prototyping (SE, UE) and evaluation (UE), therefore also means more flexibility in choosing, respectively in changing the IT supplier. As a client, Daimler AG can build up own competencies in specifying corporate systems through employing appropriate tools during RE. Fig.1 illustrates what the optimized process for corporate software development at Daimler AG looks like. After an initial and usual claims analysis, the back- and front-end of the system can be modelled separately and in parallel with specific tool support. This course of action demands close cooperation and intense communication between developers, but enables a cost- and time-efficient prototyping. This provides rapid feedback and guides the overall specification process towards an ultimate design solution. The UI design is iteratively tested, incrementally enhanced and finally accepted and forwarded as a living UI specification.

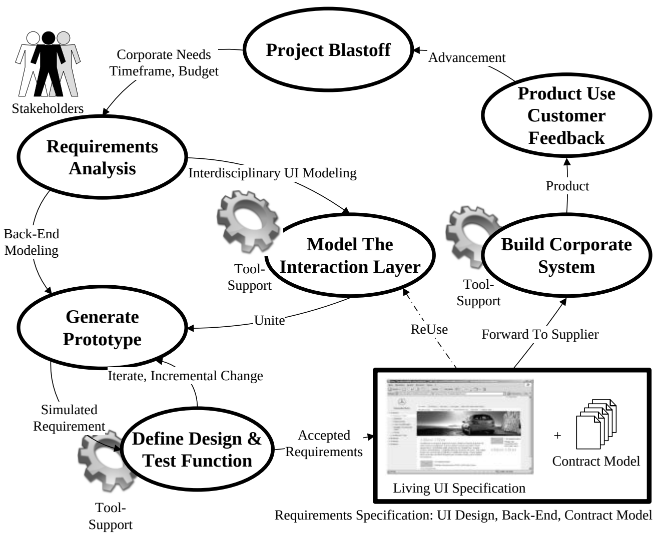

Figure 1: Interactive UI specification process

As outlined in Table 1, there are several differences between UI prototyping and the interactive UI specifications as we propose them. Whenever the IT supplier needs guidance on how the system must look and feel like, one can pop-up the simulation and easily build the corporate software system accordingly. Integrated information, accessible at both the UI and in the underlying models, make the UI design rationale transparent for stakeholders. Paper-based documents become less important and the overall process more relying on expressive interactive representations.

Table 1: UI prototypes vs. UI specifications

UI Prototypes	UI Specifications
Vehicle for requirements analysis	Vehicle for requirements specification
May be inconsistent with text-based specification and other graphical notations	Integrated explanations enable tracing the process of translating claims into the UI and vice versa.
Cheap when abstract, expensive when detailed	More initial effort, less effort for generic changes
Either low-fidelity or high-fidelity	Flexible fidelity due to segregation of models
Supplements text-based specification	Substitutes text-based UI specification
Design rationale saved in supporting documents	Incorporates design knowledge and rationale

3. Tool support for UI specification

Taking into account demanding conditions for development (see Section 1), we had an eye on agile principles and practice for providing adequate tool support [1,2]. This mainly concerns reusability of resulting UIs, support for rapid changes and iterative development and improved communication. As described before, the overall requirements for a UI specification tool derive from web projects at Daimler AG and are summarized as follows: (1) the interactive UI specification substitutes for office documents and functions as a vehicle for discussions, (2) the underlying formal UI description language allows an easy generation of alternate UI design solutions, (3) the implementation of different versions of a UI should be easy and fast, such that changes can be done on the fly during focus groups, (4) the early externalization of design vision allows an up-front usability evaluation of look and feel before system implementation, (5) the early detection of UI and usability issues prevents costly late-cycle changes and helps to keep an agile timeframe, (6) the prototype can be forwarded as a visual, interactive UI specification to a supplier.

3.1 Related work

With regards to a previous analysis of tools for RE and their applicability for UI specification in [1], we agree that existing tools do not satisfy our requirements. Typical web development suites such as *Adobe Dreamweaver*[1] or *NetObjects*[2] help in designing the UI, but advanced designs cannot be created without coding. Then, besides HTML, other languages such as Cascading Style Sheets (CSS) or JavaScript must be known by the UI designer. Furthermore, such environments do not allow the UI design to be independent from a specific technology and runtime environment. The more sophisticated tools *iRise Studio*[3] and *Axure Pro*[4] successfully make coding widely unnecessary. But their output formats are again either limited or proprietary. And, concerning our problem domains at Daimler AG, we typically deal with very complex information spaces (see Section 4). In many cases, we primarily need to rapidly evaluate appropriate information architectures and navigation structures that most suitably provide access to content. Existing tools mainly focus on the design of individual pages with static links and barley have an eye on the big picture, not to mention a pipe of changes at the navigation model to all dependent pages. We therefore decided to build our own tool at Daimler AG, since a domain specific language (DSL) is a better fit with our problem domains than commercial tools could be.

3.2 Tool architecture

In accordance with the above-listed demands on the tool, we developed MAXpro (MAX points to modeling, agile, XML) such that it can be employed by an ergonomist with basic background in XML. Such experts are typically involved in our projects where the UI has great weight and they are capable of using sophisticated tools. Because of the DSL (derived from XML and XSL), programming is unnecessary for creating interactive simulations.

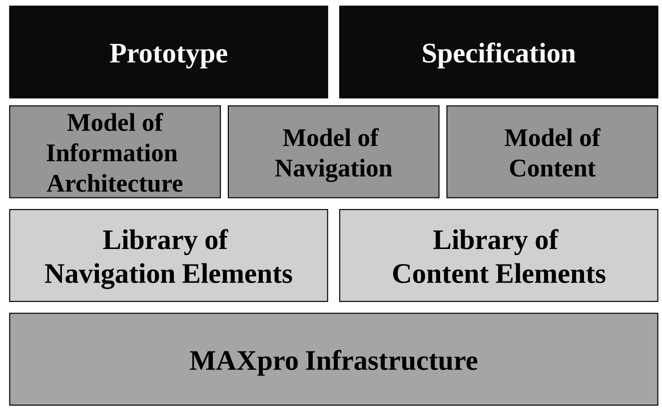

Figure 2: MAXpro architecture overview

Through a strict disjunction of content, navigation and information architecture model (see Fig. 2), a domain expert can focus on modeling content pages before thinking about their concrete place in the overall information space of the website. Conversely, one can model the information architecture before the detailed content is available. The interconnected structure of pages is modelled independently of the actual navigation concept. According to his separation, we also distinguish three different main types of XML documents for generating prototypes and the final UI specification. Libraries with navigation and content elements will later be used to fill in widgets and design. These can be of varying detail, such that switching between low- and high-fidelity UI design becomes easy (see Table 1). Moreover, the libraries can be designed for different systems and software architectures, what makes MAXpro a potential cross-platform tool.

Which elements of the information space are to be linked to specific parts of website navigation is defined through *bindings* in XML. Independent, but interrelated parts (e.g. having own navigation structure) are summarized as *modules* (see Fig. 3). As the dynamic generation of the prototype is based on these bindings, changes to a single page and its position in the hierarchy or its integration to navigation modules are forwarded generically. Additional links that do not automatically emerge from modeling the information and navigation structure can be defined in a *linkset*. *Pages* can have different *views*, allowing content modeling of even quite complex pages and a faster visualization of design alternatives. Furthermore, pages can consist of subsets of other pages. The overall set of all pages, bindings and modules is the *application*. Knowing various kinds of dynamic structures, MAXpro is able to model various kinds of information architectures, reaching from simple linear structures to complex tree and networked structures. We use generic, dynamic structures to model e.g. pop-up dialogues or contextual links that lead into completely different parts of the information landscape (e.g. disjoint subtrees).

[1] Adobe Dreamweaver, http://www.adobe.com
[2] NetObjects Fusion, http://www.netobjects.de
[3] iRise Studio Suite, http:// www.irise.com
[4] Axure Pro, http://www.axure.com/

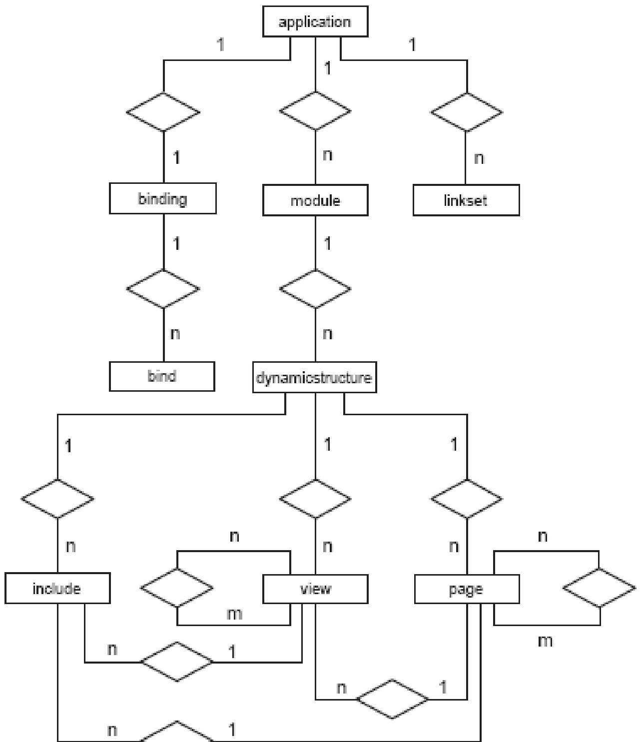

Figure 3: MAXpro architecture (outline)

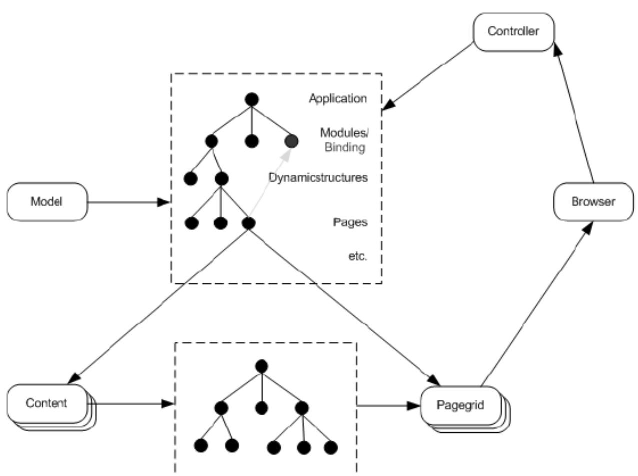

Figure 4: Prototype generation process

Furthermore, it may be necessary to break up the structure in smaller parts in order to be able to handle complexity. Such functionality also allows reuse of specific, component parts of the structure and reduces redundancy in the XML description. For compiling the prototype, we use XSL(T), JAVA and JDOM to transform formal XML descriptions of the UI into running code (e.g. HTML or JSP). Altogether, MAXpro allows the generation of two different types of prototypes, namely (1) a static copy of all generated and connected pages and (2) a dynamic prototype which is built at runtime. The static version can be run in a web browser without installing other components. It is therefore especially suited for being forwarded to stakeholders for evaluation and to suppliers for implementation. If the MAXpro runtime environment is installed on the client machine as well, one can also generate prototypes at runtime. The advantage then is that changes to the XML source files can be immediately compiled to a running UI simulation: if the user opens a web page of the prototype, the underlying models are (re-)computed and the site is (re-)built (see Fig. 4). Due to the information in the *navigation* and *content* model, the system knows how the current *page*, depending on its specific position and state in the information space, has to be displayed.

4. Mercedes-Benz application scenario

During the development of the Mercedes-Benz website, we used MAXpro for early abstract UI design prototyping (see Fig. 5), discussions with stakeholders, UI evaluation and finally for detailed visual prototyping to deliver an UI specification to suppliers (see Fig. 6). For modeling the UI, the ergonomist employs MAXpro to structure the content of the site into an information architecture using a simple XML file. Allocating pages with their IDs in a hierarchy makes up the navigation model for the information space. Bindings map the different levels of the hierarchy to navigation widgets. This action typically takes place before the detailed content is defined by marketing teams. Accordingly, single pages will at first integrate the according navigation structure and represent it using very simple widgets (e.g. lists) accompanied by blind text. The look and feel of the widgets is defined in XSL. The early prototype is used to evaluate and discuss the basic structure and page flow of the website for later refinement. Meanwhile, the designer is employed with preparing high-fidelity navigation controls and templates for detailed UI design. Finally, the overall Mercedes-Benz web style guide is imported, resulting in 15 different layout templates.

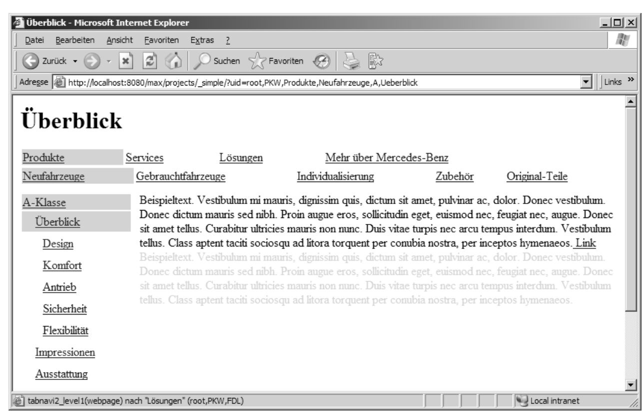

Figure 5: Abstract UI prototype for Mercedes-Benz

All content elements (e.g. text boxes, teaser images, headline types etc.) and navigation modules (e.g. breadcrumbs, tabs, pull-down menus, jump menus etc.) are modelled by domain experts and prepared for usage within the high-fidelity prototype. As soon as the information architecture is consolidated, the ergonomist links pages to detailed content and hence defines the detailed look and feel of the web pages. The page templates are divided into components that together make up the UI. They can also

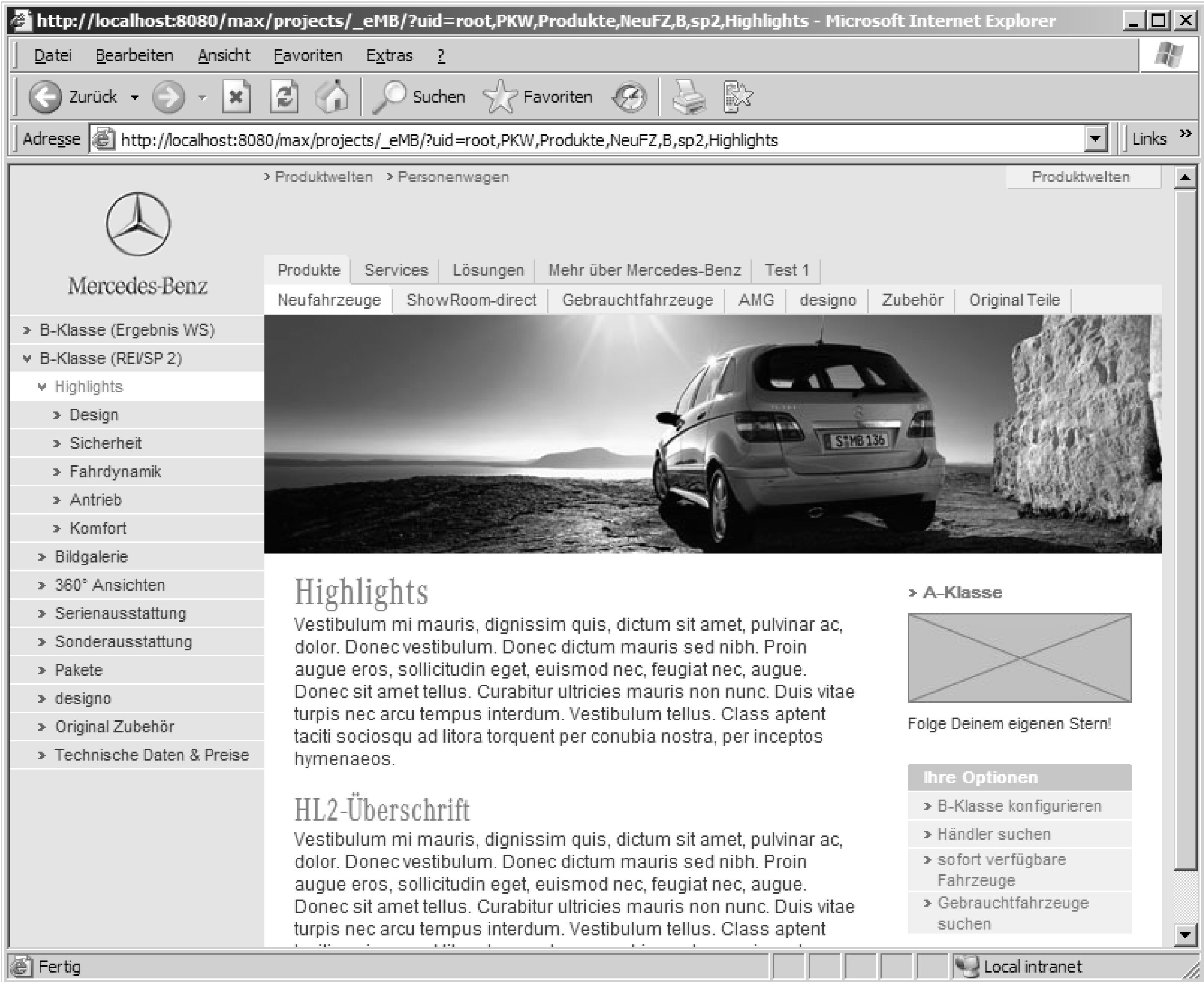

Figure 6: Detailed UI specification prototype

include e.g. JavaScript, Flash or other dynamic elements to determine the interactive behaviour of the interactive UI specification. Simple controls are replaced with designed ones, content is enriched by more precise text and images. Due to the segregation of models, one can e.g. exchange the contents or navigation concept to try out different versions of the UI. This is especially useful for determining the interactive behaviour that is most consistent with CI and CD.

5. Lessons learned

Since we began to extensively employ MAXpro for interactive system specification, we have gathered varying experiences with our tool approach. Within the Mercedes-Benz website project we found that using MAXpro was of great advantage in narrowing the design space to a solution that best addressed the requirements of stakeholders. Instead of just relying on text-based descriptions of important parts of the UI, complex user tasks could at once be modelled and visualized in varying fashion. This helped to understand the pros and cons of many ideas and also increased the understanding of involved IT suppliers that usually need much time understand industry-specific problems. We could thereby reduce the amount of workshops and we agreed upon design solutions more easily. Through the early visualization of the UI, stakeholders unable of abstract thinking in matters of IT could participate in decision making from the very beginning. This significantly reduced the amount of costly change requests due to misunderstood or incomplete requirements.

Achievements

- Vehicle for communication and discussion
- Stakeholders can speak a common language
- Iterations between discussion, decision and redesign are shortened or even eliminated
- Early evaluations of non-static UIs are possible and prevent costly late-cycle changes

- The development team can build rapid abstract and detailed prototypes rapidly
- Navigation concepts can be implemented and assessed quickly
- The visual simulation can be forwarded to the supplier as living UI specification
- Having its own UID competency, one is less dependent on specific suppliers

Shortcomings

- High initial effort to create DSL
- Stakeholders cannot pro-actively take part in participatory design without knowing the DSL
- Contextual information about design decisions and design rationale is only provided as supplementary text
- Defining contracts becomes a challenge due to the linkage to detailed specifications

Naturally, MAXpro is not yet a perfect fit to all project variables. Regarding the forming of contracts, UI specifications have to be applied reasonably and thoughtfully. On the one hand, the development of extraordinary detailed specifications on client-side is too expensive. On the other hand, suppliers have to reduce their risk of engineering failure through strict translation of specifications into code. This trade-off has to be well balanced. For example, it should not be necessary to redundantly specify the design of e.g. forms or widgets, if they appear in slightly different versions. Otherwise the work load of an interactive UI specification would easily reach that of implementing the final system. MAXpro helps to substitute PowerPoint slides and mockups, simulates look and feel at low- and high-fidelity and functions as a pilot system due to the transformation of XML into code. But, for non-technicians, a critical shortcoming of MAXpro is the necessity of understanding XML for building simulations. For covering this topic, we envisage a GUI designer as a supplement. Then, any stakeholder can pro-actively take part in the UI design process. A similar approach is committed by the usiXML editor(s) [4]. But at Daimler AG, the challenge will be to provide a GUI builder regarding the customs of stakeholders in using Office applications. Furthermore, MAXpro only specifies the UI of a website. Although being inappropriate for corporate UID, abstract models are as well important for a complete UI specification. We found that is not always sufficient to integrate information about users, their tasks and related information as text. As with the UI, MAXpro needs to include typical RE models in a visual manner as discussed in [1,2]. In order to understand all requirements and for being able to back-trace the origin of an UI design, some stakeholders demand business cases, user role maps and UML diagrams. This is e.g. necessary to communicate in detail with technicians and the ones in charge of back-end design. The smooth conversion between different kinds of models and the UI would provide additional bridging between the disciplines of stakeholders [1]. Consequently, for communication and a shared understanding, all models must be accessible from the UI layer. We are also considering the provision of an annotation function. This will be helpful for design assessment and user feedback on both UI and underlying models.

6. Conclusion

Interactive UI specifications are especially interesting for corporate software development, because actual system implementation is mostly outsourced. This frequently causes a loss of control on UI design. Hence, the UI must be specified at an early stage in order to ensure compliance with CI and CD. Prototypes have to become the visual spearhead of a shared understanding about look and feel. With MAXpro, we are able to build prototypes of different fidelities, to evaluate them at an early stage and to deliver them as detailed UI specifications to our suppliers. Altogether, this makes corporate UI development a controllable and traceable process with many advantages for Daimler AG (see Section 5). Consequently, we already apply MAXpro in complex web-based application domains, where navigation structures need to be thoughtfully tested. Using a problem-adequate DSL much better fits our requirements as other tools or XML-based languages like usiXML [4]. Due to other experience in applying DSLs for UI specification [2], we promote our approach as being valuable for UI specification processes similar to the one at Daimler AG. We will continue to enhance and extend our UI specification tool in order to be able to comprehensively support all stakeholders during all stages of the process. We envisage a consistent tool-support from basic requirements analysis and models to visual UI specification. Due to the separation of models and the chance of providing different UI libraries, we also consider using MAXpro for the specification of desktop applications and in-car information systems.

References

[1] T. Memmel, F. Gundelsweiler, H. Reiterer, Prototyping Corporate User Interfaces – Towards A Visual Specification Of Interactive Systems, *Proc. of the IASTED-HCI 2007*, Chamonix, France, 2007, 177-182

[2] T. Memmel, C. Bock, H. Reiterer, Model-driven prototyping for corporate software specification, *Proc. of the Engineering Interactive Systems (EIS) 2007*, Salamanca, Spain, 2007; available online at http://www.se-hci.org

[3] I. Horrocks, *Constructing the user interface with statecharts (*Addison-Wesley, Harlow, 1999)

[4] I. Pederiva, J. Vanderdonckt, S. España, I. Panach, O. Pastor, The Beautification Process in Model-Driven Engineering of User Interfaces, *Proc. of 11th IFIP TC 13 Int. Conf. on Human-Computer Interaction INTERACT'2007,* Rio de Janeiro, Lecture Notes in Computer Science, Vol. 4662, Springer-Verlag, Berlin, 2007, 409-422