Applying the MVC Pattern to Generated User Interfaces with a Focus on Audio

Dirk Schnelle and Tobias Klug

Telecooperation Group
Darmstadt University of Technology
Hochschulstrasse 10
D-64283 Darmstadt, Germany
{dirk,klug}@tk.informatik.tu-darmstadt.de

Abstract. Mobile users can interact with devices in the environment either by operating them directly or through personal devices carried by the users. This requires an adaption of the user interface to the device used. Declarative markup languages are considered to be a solution for single authoring user interfaces for different devices and modalities. This is a challenging task, since each device has its own characteristics. We present in this paper a novel architecture to support the creation of user interfaces based on a declarative markup language and a UI-independent task model. This architecture is based on the Model-View-Controller pattern (MVC) to generate user interfaces from declarative markup languages. We introduce a clear border between a modality independent task model and UI design. We also show how the development of smart environments can benefit from the workflow engine underlying our architecture.

1 Introduction

Ubiquitous computing poses new challenges to human-computer interfaces. One of them is the use of input devices that are available in the environment or which the user carries with her. As a consequence it is necessary to adapt the interaction with the user to the current device. This implies also the use of different communication channels, or *modalities*, which are used to interact with the user [3]. The development of multimodal applications is complex and time-consuming, since each modality has unique characteristics [12]. Research and modern development approaches try to solve this by means of declarative markup languages, mostly XML-based.

They promise that the same model can be reused for different modalities and output devices [20]. The reusability concerns mainly the so called *business logic* of the application. We understand business logic as a specification of the concept: Who (roles) does what (task) when (process) and how (environment). The main requirements in business logic are reusability and the independence of the user interface. But the community of authors and users of generated

C. Stephanidis and M. Pieper (Eds.): ERCIM UI4ALL Ws 2006, LNCS 4397, pp. 192–210, 2007.

interfaces have already discovered the limits of this approach. “Using the same user interface for all devices means that the thinnest device will set the limits for the user interface, and unless the user interface is extremely simple, some device categories necessarily will be excluded” [15]. Gerd Herzog et al. [8] come to the same conclusion that “it is impossible to exclude all kinds of meaningless data from the language and the design of an interface specification will always be a sort of compromise” [8].

One of the modalities for which such a compromise has to be found is audio. Its invisible and transient nature poses many challenges to the interface designer [19]. Due to the fact, that multimodal applications are often considered to have both, graphical and audio in- and output capabilities, it is astonishing, that audio-only applications seem to be considered with lower priority.

Lingam [10] first seems to ignore this fact by stating that the “[. . .] day the limitations of voice recognition and Natural Language processors are overcome [. . .]” [10], but he also argues that not all limitations can be overcome. He sees a solution in a complementary use of all available modalities. This is also the common tenor, that voice can only be used as a complementary modality, but not on it’s own. Because humans are more visually-oriented than aural, there is much more research being done with a focus on graphical rendering, i.e. layout of forms, than on audio-based interfaces.

Shneiderman names the example of a stock market where a survey result showed, that although trading is done by voice, the visual approach is 10 times more attractive to users [19]. The limits of audio are not well understood and are therefore replaced by the approach to use audio as a complementary modality. However, under some circumstances audio is a first class medium. Especially for visually impaired people or for workers who do not have their hands and eyes free to interact with the computer.

In this paper, we introduce a novel architecture to support the creation of user interfaces based on a declarative markup language and a UI-independent task model. This architecture satisfies all requirements in a reusable business logic and combines it with the advantages of generated user interfaces.

The architecture applies the MVC pattern to generated user interfaces using existing standards i.e. workflow engines and XHTML. Our main goal is to remove all implementation details from the model by separating the application into a modality- or implementation-independent part executed by the workflow engine and a modality-dependent part executed by a renderer for each supported modality. Using a workflow engine as the central building block allows us to easily integrate backend systems, user interfaces and other services like sensors and actuators to the environment via a standardized API.

The rest of this paper is structured as follows. In the motivation section we name shortcomings of existing solutions for UI generation. The next chapter takes a closer look at existing comparable solutions. Then we give an overview about our architecture and the way from the task model to the UI in section 4. Two use case scenarios for voice-based user interfaces and graphical user interfaces are given in section 5 to demonstrate how our approach can be applied to