Chapter 13

A Global Process for Model-Driven Approaches in User Interface Design

Sybille Caffiau
University Joseph Fourier – Grenoble, France

Patrick Girard
University of Poitiers, France

ABSTRACT

In user interface design, model-driven approaches usually involve generative solutions, producing interface by successive transformations of a set of initial models. These approaches have obvious limitations, especially for advanced user interfaces. Moreover, top-down design approaches (as generative approaches are) are not appropriate for interactive application development in which users need to be included in the whole design process. Based on strong associations between task models and dialogue models, the authors propose a global process, which facilitates the design of interactive applications conforming to their models, including a rule-checking step. This process permits either to start from a task model or a user-defined prototype. In any case, it allows an iterative development, including iterative user modifications, in line with user-centered design standards.

INTRODUCTION

Since the 70’s, application designers and developers have tried to define and to structure the design steps in order to improve application development. Design models were first defined (Winston, 1970).

DOI: 10.4018/978-1-4666-1628-8.ch013

With the popularization of the use of computerized systems (and then, the proliferation of interactive applications), they looked for integrate users in design modeling. Design models were adapted (such in the V model) and new models were created. For example, the star model (Hix, 1993) proposed a non-sequential design process centered on the evaluation step. This design model introduced
iteration and user feedbacks in the interactive application design process.

Design models identified design steps and links between steps, however, they did not define link semantics, and thus, they did not explain how to go to the next step from previous ones. Several works proposed model-based approaches. They developed concrete interface from abstract interface definition by combining and transforming models (Calvary, 2003).

In such model-based approaches, task models stand for the user point of view. Task model express activity as a hierarchical structure, based on the “theory of action” (Norman, 1990), which describes actions as compositions of sub-actions. The task model is often the only entry for user needs and observations. Thus, several research works (Mori, 2004; Luyten, 2003; Wolff, 2009) use a generative approach to build user interfaces—mainly skeletons to be completed—from task models. Following the analysis we made in Caffiau (2007), we can argue that this approach has several limitations.

First, generating requires adding information in order to reach interface operative stage. This information can be added to high-level models, which then loose their original goal; so doing they become hard to understand and to use, because of their multiple semantics. For example, task models use operators that schedule tasks. Associating tasks and interface components (widgets), the interface components included in the interface presentation may be deduced. Nevertheless, when tasks can be concurrently performed, there is no way to infer the position of the interface components. For instance, in a form, the order of text fields (and their associated labels) cannot be inferred from any initial model. The way generative approaches work is to affect a position to the components that depends on the place the corresponding task have in the task tree (i.e. if, in a task tree, a task entering addressee is placed left to a task entering message, in a mailer form, the text field used to edit the addressee will be placed above the text field used to edit the message). Then, presentation information is added to task models through the addition of new semantics to this model. The other way is to insert this information during the generating process. This second approach is for example used in TERESA (Berti, 2004) by the way of heuristics, which are applied during the process. This however results in a lack of understanding of such transformations by users.

Second, all considered research issues are concerned with classical WIMP applications. The hierarchical structure of task models is used to build the interface navigation scheme. We demonstrated in Caffiau (2007) that introducing non menu-based interactions implies a non-automatic transformation of the application dynamic expression (named dialogue).

Last, generating is not easy to include in iterative design cycles such as HCI-adapted cycles. When changes are required, it is necessary to modify the high-level models, and to generate again a new skeleton, to be improved again by hand-made add-ons. Some results have been obtained around the definition of “round-trip engineering” (Hettel, 2008; Sendall, 2004), but have not been applied to HCI yet. More, these approaches prevent the designers to start from the prototype, which method is often used in post-WIMP design.

Our aim is to introduce a new way to use models in user interface design. Our approach proposes to link some concepts of two models (task model and dialogue model) in order to design and check the application dynamics). Leaning on metamodels of one task model (K-MAD [Lucquiaud, 2005; Baron, 2006]) and one dialogue model (HI [Depaulis, 2002; Depaulis, 2006]), we wrote equivalence rules between such models. Then, we defined a new development cycle that can be used in a user-centered iterative approach.

We present the global approach in the first section. It is based on links existing between concepts of the K-MAD task model and the HI dialogue model, which are relevant for the applica-