

Issues in Automatic Generation of User Interfaces in Model-Based Systems

Angel Puerta

List of Participants

Mark Addison (Phillips Research, UK), Bernhard Bauer (TU Munich, Germany), Javier Contreras (UAM, Madrid), Martin Fischer (University of Bonn, Germany), Fernando Gamboa (INRIA, France), Frank Hofmann (University of Bochum, Germany), Volker Kruschinski (University of Bochum, Germany), Frank Lonczewski (TU Munich, Germany), Angel Puerta (University of Stanford, USA), Siegfried Schreiber (TU Munich, Germany), Jean-Claude Tarby (U. Lille, France), Pedro Szekely (ISI, University of Southern California, USA)

Introduction

The working group examined the issue of what level of automation is desirable, or effective, in interface development, especially in model-based systems. Two camps emerged that were very much apart at the beginning and made small concessions towards the end. One camp advocated using no automation at all, instead letting interface developers make design decisions, perhaps with decision support from a system. The second camp proposed that maximum automation of interface design should be the goal of a model-based systems.

The Positions

These are the arguments made by researchers who favour automation in interface development:

- *Well-defined interface design processes are feasible.* It is possible to develop methodologies and theories that establish processes for completing interface designs. For example, it has been shown that generation of layouts from data models is fast and efficient in a fully-automated way.
- *No conceptual problems, just technical ones.* Any limitation currently faced by MB-IDEs is due to the lack of appropriate methodologies for automation, which can be eventually developed, and not due to inherent technical barriers.

- *Facilitates rapid prototyping.* The gains in rapid prototyping via automation cannot be duplicated by model-based systems that offer only developer support.
- *It's cheaper even if it is not better.* The saving in resources for development in automated systems clearly outweighs the loss in quality and flexibility inherent to automation.

In contrast, those opposing automation in development environments for interfaces put forth these points:

- *Interface design knowledge is a moving target.* One of the basic problems with model-based systems is that by the time that current interface design knowledge has been coded into the system, such knowledge is obsolete.
- *Knowledge representations are too complex.* We have not defined an efficient way to represent interface design knowledge. Most methods that have been used do not scale up well, or cannot be generalised.
- *Developers don't want generated interfaces.* One of the most pressing practical barriers to automation is that more often than not developers are not happy with the generated product. This creates a need for customisation that automated systems cannot efficiently provide due to its own nature.

The Points

Group members were able to coincide in a number of points:

- *Minimum input.* Any model-based system, regardless of the intended level of automation must count with a minimum input. This input normally consists of user-task and domain model information. This fact is a reflection that the field does not have a good understanding of the methodologies or theories that could be used to generate such information from other input formats.
- *Automated generation must be studied by model component.* The level of automation must be examined component by component of an interface model. Thus, within each components it is possible to identify subprocesses where automation is agreed to be desirable.
- *Tool support is not automated generation.* It is important to distinguish these terms. Support consists of any tool or set of tools that allow developers to define parts of an interface model. Automation consists of any tool or set of tools that produces a part of an interface model based on another part of the same (or another) model.
- *Automation possible is inversely proportional to abstraction level.* This seems clear but it is worth emphasising. Abstract objects such as user tasks are much more improbable targets of automation than, say, the layout of widgets.

The Conclusions

- *Systems must have an automation «knob».* The idea here is that while automation vs. support may be a lively debate, users of model-based systems should not have to be limited to just one side of the issue. It is important, therefore, that model-based systems offer the capabilities that would allow developers to control, to a certain extent, the degree of automation that the system provides. Some developers may opt for fully automated interface production whereas others may opt for manual design, or more likely, a mixed approach.



Figure 1. Desirable automation knobs would have separate settings for each component of an interface model

- *Knobs must exist for each model component.* Because of the various points raised above, a single knob would not do the job. Each model component, and each design process, or group of processes, should be adjustable for automation.
- *Current systems do not have any knobs.* It is quite clear that no model-based system either constructed or under construction offers much in the way of adjustable knobs. This should be an immediate goal in systems under development
- *Automated generation has only been proven in narrow application domains.* There is no evidence that automated generation of interfaces can be extended beyond the restricted application spaces that have been examined already. This may be an inherent limitation.