

Reflections on Model-Based Automated Generation of User Interfaces

Angel R. Puerta

The main contribution of Mecano was to incorporate the use of domain models, or ontologies, into the model-based interface development framework. Paradoxically, it was this same feature that unveiled the limitations of the Mecano approach. Domain models allowed us to generate not only the static layout, but also significant portions of the dialog of an interface. This was possible thanks to the much greater expressiveness of a domain ontology over a traditional data model. Many groups had employed data models before to support automated generation of interfaces and that limited them to producing static layouts only. Mecano was used to build a number of form- and graph-based medical and military interfaces.

Soon, however, after we embarked on the ambitious project of generalizing the Mecano approach beyond forms, we found ourselves hitting the proverbial wall. We reasoned that our interface models must be comprehensive, go beyond domain representations and incorporate all relevant aspects of an interface design, including user tasks, presentations, dialogs and users. In this we succeeded by creating a new modeling language for interfaces. We supported that language with an open client/server architecture that would allow multiple tools and environments to communicate with, operate on, and edit the interface model.

But the design space for interfaces exploded under this framework. The computational force and knowledge bases required to automate the design of interfaces, in the ample sense of the word, was unattainable. We questioned then the whole purpose of an environment like Mecano with its drive towards automating significant portions of the design process for an interface. The result was a new, redirected effort to build a model-based development environment with the same aim of Mecano—developing interfaces from a declarative model—but with a significantly different philosophy. Instead of automation, we looked for ways to support interface designers. Instead of large knowledge bases and complex inferencing strategies to make design decisions, we built interactive tools for developers to do what they do best: design interfaces. Instead of using interface models as a driving mechanism for an expert system, we used them to organize and visualize interface design knowledge.

Our successor to Mecano, called MOBI-D (Model-Based Interface Designer) (Puerta and Mulsby, 1997; Puerta 1997), is now a much different product. MOBI-D supports a user-centered design methodology that involves end-users in the design process. It is task-oriented (as opposed to domain-oriented), and is an interactive environment, not an automatic one. Of course, we support the automation of menial or tedious tasks, but we do not try to take control away from designers. This is certainly a marked departure from previous efforts in model-based development, a new philosophy that other groups around the world are now starting to adopt. Thus, Mecano showed us the power of added expressiveness in an interface model. It also showed us the limits of automation and pointed us to a new path and a new paradigm.

References

Puerta, A. 1997. Model-Based Development Tools. *IEEE Software*, 14(4), July/August 1997, pp. 41-47.

Puerta, A. and Maulsby, D. 1997. MOBI-D: A Model-Based Development Environment for User-Centered Design. In *Proceedings of CHI'97* (Extended Abstracts), Atlanta, March, pp. 4-5.