An EUD Approach for Making MBUI Practical

José A. Macías and Pablo Castells
Universidad Autónoma de Madrid
Ctra. De Colmenar Viejo, Km. 15
28049 Madrid - Spain
+ 34 91 497 {2241, 2284}
{j.macias, pablo.castells}@uam.es

ABSTRACT

In this paper, we present our perspective on Model-Based User Interfaces (hereafter MBUI) paradigm and provide with our experience in this area combining high-level knowledge-based data models (i.e. ontologies) and reverse engineering processes to carry through a pragmatic MBUI vision. Our approach is based on using End-User Development (hereafter EUD) techniques (i.e. Programming by Example) to enable the user to carry out editing tasks in a MBUI environment. This advocates an EUD-for-MBUI approach, where the system avoids the user from having to deal with interface specification languages.

Keywords

Model-Based User Interfaces, End-User Development, Programming By Demonstration, Knowledge Representation, Semantic Web, Web Authoring Tools.

INTRODUCTION

Roughly a decade ago, the automatic generation of user interfaces became a first order concern for even the simplest software application. As the process of generating automatically the interface involves a certain effort, MBUI emerged as a new approach that claimed to overcome several difficulties in automating the process of generating interfaces (i.e. redundancy, lack of encapsulation and reusability). This advocates the idea of splitting up the conceptual level of an interface, which leads to the explicit specification of different aspects of a user interface, such as knowledge, presentation, dialog and behavior.

There are several features of MBUI approach that makes it appealing from the designer point of view. One of them is probably the explicit separation into different models that represent the conceptual levels of the interface design. Such a characteristic helps meet reverse-engineering processes [12] that bring promising new approaches. This way changes can be made to the user interface by means of the implicit reverse path that comes from the generated version of the interface to the underlying models.

All in all, MBUI approach suffers from implicit lacks that might make this approach likely to be considered useless, cumbersome and, in general terms, hard to exploit. To this respect, probably one of the main problems of MBUI is about dealing with the languages used for the interface specification. In this sense, a great deal of users and IT professionals might consider MBUI specification languages as hard to use as to codify in ordinary programming languages. To solve this lack and make easy the way users tackle such languages, several approaches appeared in order to provide a set of design tools intended to design different features of a user interface [8]. These tools relieve the user from having to manipulate the specification language(s) and allow high-level specification capacities. However, there is an implicit problem with all this design tools. Since the user has to use an abstract specification language, s/he also has to stick to the visual language in order to deploy the design tool. Conceptual models and abstracts have to be kept in mind while the user designs the interface, making difficult the design process since the user does not has to be a programming expert but rather be a domain-expert.

Another problem, probably more related to IT professional necessities, is the non-standard characteristics of most interface specification languages. Nowadays, there is a real necessity of adapting applications for standardization and knowledge sharing. The semantic web claims to carry out such an approach, using XML-based languages as a distribution and specification medium for a great deal of services and (web) interface construction.

SEMANTIC SPECIFICATION

Few years ago, the semantic web paradigm proposed a new challenge for knowledge representation and web automation. New languages in turn appeared for dealing with complex relationships among contents to be dynamically generated and finally represented in a web browser.

We firmly believe in ontologies as a medium to specify different features of interfaces and, specifically, web-based interfaces. Ontologies provide a conceptual model where complex relationships can be defined in order to codify high-level semantic path for further characterization and

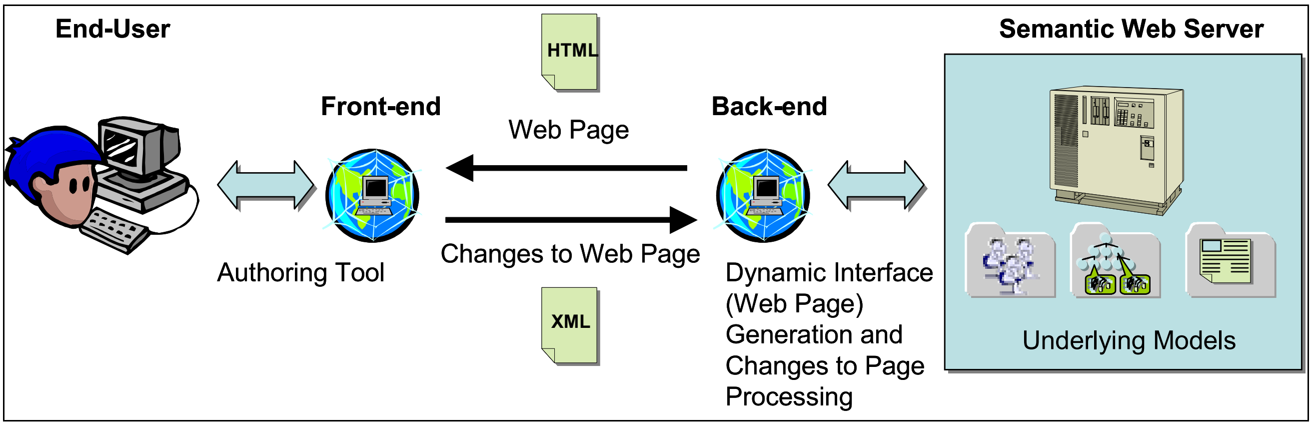

Figure 1. Our MBUI approach

reverse-engineering purposes [5], [7]. Our research experience is in using ontologies as a medium to specify knowledge for building data models (i.e. domain models) used together with application or presentation models. This makes possible for our approach to specify complex knowledge focused on the interface's domain model and also work with XML-based languages that better fulfils the above assumptions about distribution and sharing.

Since semantic web languages have growth in popularity, different specification languages appeared in order for the designers to model complex multi-modal user interfaces. Examples of such languages are UIML [11] and XIML [9]. New technologies bring semantic web services that provide distributed dialog models intended to codify complex relationships between application servers. These approaches are garnering a great deal of popularity among IT professional, helping meet new strategies based on XML to enhance and automate the web as well as the web interface functionality.

All in all, ontology-based systems remain the same problem as classical MBUI approaches: how to easily deal with the specification language?

EUD FOR MBUI

One of the most important matters we have been dealing with during these years is making easy the way the user interacts with computer in her/his daily problem solving activities [6]. This could probably be a good starting point that can help bring MBUI approaches and IT professionals together.

End-User Development techniques allow for the *Gentle Slope of Complexity* and make easy the way the user accomplishes tasks by means of computers [3], featuring well-known paradigms such as Programming by Demonstration [2], [4].

EUD techniques can be applied to MBUI in order to relieve the user from having to deal with languages focused on interface specification. Sometimes this implies to reduce the expressiveness of MBUI in a certain way since the user might not need to manipulate declarative specifications of the interface any longer. To carry out this trade-off between expressiveness and complexity, it is necessary for the system to provide with a low-level abstract design environment, somehow close tie to WYSIWYG approaches that provide the end-user with a real representation of the interface aspect. In such environments the user can easily manipulate the interface's objects rather than using complex visual or specification languages. This way it is possible to provide with a more accurately vision of what is attempting to do at every step. However, creating an interface from the scratch by means of WYSIWYG environments is not always such an easy task, since a lot of implicit information from the underlying models is often required [7].

MAKING MBUI EASY

Our approach aims at combining MBUI systems and Programming by Example techniques that help make changes to explicit interface models automatically. Therefore we have been dealing with web-based user interfaces generated automatically (i.e. dynamically generated web pages) by using ontologies as explicit language for the web-based interface specification [6].

Figure 1 depicts how our approach works. On the right side, a Semantic Web Server generates web pages by means of interface models, that is a domain model for representing contents, a presentation model that specifies the layout and style of the objects being presented and a user model in order to adapt the contents of the page to a specific user profile. The page is generated and is in turn sent to the front-end. Next, the end-user receives the web page and navigates through it. The user can also make changes to the web page, such changes are sent back to the server and the back-end application attempts to analyze this information and infers what models have to be modified as well as what information have to be updated into the underlying models. Next time the page is generated, the information will show updated. As mentioned before, MBUI allows for an implicit reverse engineering process than can be used, in that case, to detect changes the user achieves on a generated (web) interface to finally apply such changes to the constructor models.

The main goal of our work is to find out how far one can go without leaving the WYSIWYG approach. This might envision a more pragmatic view of MBUI paradigm, where no basic assumptions are made about the user's skills in specification languages and where the automatic process of reverse engineering enables to reduce maintaining costs and efforts.

In general terms, we believe that the user must not be aware of the internal editing and building process of the interface. Furthermore, more standard specifications should be provided in order to make MBUI paradigm more transportable and shareable according to new technologies. On the other hand, MBUI makes possible reverse engineering processes that could be exploited for deploying the automatic use of MBUI paradigm itself, this process being as transparent as possible for the final user.

OUR EXPERIENCE

Our experience is in PBE systems that help the user editing ontology-based dynamic web pages. Actually we have developed an authoring tool, namely DESK [6], [7] that assists the user to carry out changes to dynamic web pages. DESK infers the user's intents by monitoring the user's activity and extracts meaningful information from the web page. This information is then used to get semantic relationships between the user's actions and the interface information (i.e. HTML page).

DESK is intended to reduce the gentle slope of complexity, supplying with a WYSIWYG easy-to-use user interface but, in contrast, featuring some expressive limitations since DESK is focused on WYSIWYG representations rather than abstract ones.

DESK has been used in several scenarios, and an empirical evaluation has been achieved in order to assess the usability of the authoring tool taking into account the user's opinion.

We also have previous experience in developing PBE systems such as HandsOn [1] that uses explicit information from a data model for detecting relationships between visual and textual components. HandsOn is based on the presentation model of MASTERMIND [10], where the designer can build presentation objects by means of direct manipulation in a visual environment.

ACKNOWLEDGMENTS

The work reported in this paper is being supported by the Spanish Ministry of Science and Technology (MCyT), project number TIC2002-1948.

REFERENCES

1. Castells, P., Szekely, P. and Salcher, E. Declarative Models of Presentation. Proceedings of the *International Conference on Intelligent User Interfaces* (IUI'97). January 6-9, 1997, Orlando, Florida, USA, pp. 137-144.
2. Cypher A. (ed.). Watch What I Do: Programming by Demonstration. The MIT Press, 1993.
3. Klann, M. and Fit, F. End-User Development. D1.1 Roadmap. Proceedings of the *End User Development Workshop at CHI'2003* Conference. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
4. Lieberman, H. (ed). Your Wish is my Command. Programming By Example. Morgan Kaufmann Publishers. Academic Press, USA. 2001.
5. Macías, J.A. and Castells, P. Adaptive Hypermedia Presentation Modeling for Domain Ontologies. Volume 2 of the Proceedings of the HCI International (HCII'01), *9th International Conference on Human-Computer Interaction*. New Orleans, Louisiana, USA; August 5-10, 2001. Lawrence Erlbaum Associates, Publishers. London. pp. 710-714.
6. Macías, J.A. and Castells, P. Dynamic Web Page Authoring by Example Using Ontology-Based Domain Knowledge. Proceedings of the *International Conference on Intelligent User Interfaces* (IUI'03). Miami, Florida, USA. January 12-15, 2003, pp. 133-140.
7. Macias, J.A. and Castells, P.: Using Domain Models for Data Characterization in PBE. Proceedings of the *End User Development Workshop at CHI'2003 Conference*. Ft. Lauderdale, Florida, USA. April 5-10, 2003.
8. MOBI-D web site, http://smi-web.stanford.edu/projects/mecano/mobi-d.htm
9. Puerta, A. and Eisentein, J. XIML: A Common Representation for Interaction Data. Proceedings of the *Intelligent User Interfaces Conference* (IUI'02). San Francisco, California, USA, January 13-16, 2002.
10. Szekely, P., Sukaviriya, P., Castells, P. Muthukumarasamy, J.; Salcher, E.: Declarative interface models for user interface construction tools: the MASTERMIND approach. Proceedings of the IFIP TC2/WG2.7 *Working Conference on Engineering for Human-Computer Interaction*. Yellowstone Park, USA, August, 1995, pp. 120-149.
11. UIML web site, http://www.uiml.org
12. Vanderdonckt, J., Bouillon, L. and Souchon, N. Flexible Reverse Engineering of Web Pages with VAQUISTA. Proceedings of the *IEEE 8th Working Conference on Reverse Engineering*. Stuttgart, October 2-5, 2001. IEEE Press, pp. 241-248.