Interaction Modeling at PROS Research Center

José Ignacio Panach, Nathalie Aquino, Oscar Pastor

Research Center on Software Production Methods (PROS), Universitat Politècnica de
València, Camino de Vera s/n, 46022 Valencia, Spain
{jpanach, naquino, opastor}@pros.upv.es – http://www.pros.upv.es

Abstract. This paper describes how the PROS Research Center deals with interaction in the context of a model-driven approach for the development of information systems. Interaction is specified in a conceptual model together with the structure and behavior of the system. Major achievements and current research challenges of PROS in the field of interaction modeling are presented.

1 Introduction

On the one hand, the software engineering community has provided sound models to represent the structure and behavior of software applications. The Entity-Relationship Model or UML Class Diagrams are examples of structural models, while Sequence or Activity UML Diagrams are examples of behavioral models. On the other hand, the human-computer interaction community has provided several proposals for interaction modeling, but none of them has become widely known or used. Even today, the identification of the conceptual primitives that allow interaction to be properly modeled can be considered a research challenge.

The PROS Research Center (Research Center on Software Production Methods) of the Universitat Politècnica de València, Spain, is devoted to improve traditional software production methods by providing model-driven processes to develop quality software. With this goal in mind, PROS has defined an object-oriented and model-driven development method, called OO-Method [5], which automatically generates software systems from conceptual models. PROS recognizes the importance of the interactive perspective and, therefore, sustains that the conceptual model of an information system cannot be considered to be complete after just specifying its structure and behavior. It is also necessary to specify how end-users will interact with the system. Hence, PROS proposes a holistic conceptual model in which interaction is modeled together with structural and behavioral models. In this holistic conceptual model, interaction with the end-user is represented in a Presentation Model.

2 Achievements: the Presentation Model of OO-Method

The Presentation Model [3] of OO-Method provides conceptual primitives to represent the interaction with a software application. A set of *elementary patterns* is provided to build *interaction units*. There are three basic types of interaction units that can be used to represent different interactive scenarios: 1) the execution of a method, 2) the list of the population of a class, and 3) the visualization of details of a specific object. These basic interaction units can be combined to build more complex ones. Furthermore, a *hierarchical action tree* is provided to organize the access to interaction units, defining the menu of the application.

OO-Method and its Presentation Model are supported by a commercial software suite named OlivaNOVA, which was developed by CARE Technologies (http://www.care-t.com). OlivaNOVA allows the conceptual model to be built and applies subsequent transformations until the final code of a fully functional application (structure, behavior, and interaction) is generated for different computing platforms. The OlivaNOVA tool is currently being used successfully in an industrial environment. Therefore, the Presentation Model of OO-Method can be considered a significant achievement of PROS regarding interaction modeling.

3 Current Research Challenges

Currently, PROS continues devoting efforts to the research line of interaction modeling with the aim of enhancing the expressiveness of the Presentation Model. These efforts can be grouped into two categories: approaches to generate usable systems [4] and approaches to customize user interfaces.

With regard to the first group, a method to include functional usability features [2] in any model-driven approach has been proposed. The method is based on the idea of abstracting the information contained in such usability features in order to include them in a conceptual model. Thanks to this abstraction, a model compiler can automatically generate the code for a specific programming language taking as input the conceptual model. This code supports the functionality of the usability features together with the functionality of the whole system. The proposal supports the existing tendency of addressing the treatment of usability in early stages of the software development process. With regard to the second group, a Transformation Templates [1] approach has been defined in order to diversify the kinds of user interfaces that can be generated. In this way, user interfaces can be customized according to context and user preferences before their automatic generation. This benefits the flexibility of the model-driven approach.

References

1. Aquino, N., Vanderdonckt, J., Pastor, O. Transformation Templates: Adding Flexibility to Model-Driven Engineering of User Interfaces. In Proc. of SAC 2010. ACM Press, New York, 2010, pp. 1195–1202.
2. Juristo, N., Moreno, A.M., Sánchez, M.I. Guidelines for Eliciting Usability Functionalities. IEEE Transactions on Software Engineering, Vol. 33, 2007, pp. 744-758.
3. Molina, P.J., Meliá, S., Pastor, O. JUST-UI: A User Interface Specification Model. In Proc. of CADUI 2002, Valenciennes, Francia, 2002, pp.63-74.
4. Panach, J.I., España, S., Moreno, A., Pastor, O. Dealing with Usability in Model Transformation Technologies. In Proc. of ER 2008. Springer LNCS 5231, 2008, pp. 498-511.
5. Pastor, O., Molina, J. Model-Driven Architecture in Practice. Springer, 2007.