Pattern-based approach for animating software architectures

Frédéric POURRAZ, Sorana CIMPAN, Hervé VERJUS
Lab. LISTIC- ESIA, Université de Savoie
BP 806
74016 Annecy Cedex
{cimpan,verjus}@esia.univ-savoie.fr, frederic.pourraz@etu.univ-savoie.fr
Tel. 04 50 09 65 86
Fax. 04 50 09 65 90

Keywords : software architectures, animation, pattern.

Software application design, development and evolution are crucial aspects of software engineering, especially when big size software applications have to be safe, conformant to the requirements.

The architecture centred development process places the software architecture in the core of the development process, the idea being that a maxim of "troubles" have to be detected early in the development process. The architecture is described at different abstraction levels, refinement allowing the transition from the most abstract ones to more concrete ones. The ultimate step in refinement is code generation. *Architecture Description Languages* (ADLs) and accompanying tools have been proposed during the past years [1][2][3], with the scope of supporting the architecture centred process. Two facets of architecture description can be considered: structural and behavioural. While all the ADLs offer support for the structural description, only some of them consider the behavioural aspects. The tools provided for supporting the architecture are mainly graphical modellers.

Such tools cover only the structural aspects, focusing on the static representation of the architecture. Behavioural aspects such as the architecture evolution over time or the communication between its constituent elements are not represented. These are important issues in the architecture comprehension.

In this paper we address the issue of software architecture graphical representation, covering both structural as well as behavioural aspects. In order to support both aspects our approach is to generate graphical animations for software architectures starting from their ADL based descriptions. This implies making the correspondence between two domains: the one of architecture description, and the one of graphical animations. We use a pattern-based approach in order to make this correspondence.

In the architecture description domain, we identify the aspects that can be considered for animation. In doing this we address generic issues such as encapsulation, composition, communication among elements, etc. As mentioned, we cover both structural as well as behavioural issues. Thus, we choose an ADL (π-ADL [4]) that covers both aspects for our study and we identified all the constructs of the language that have to be considered when generating the animation from the architectural description.

In the animation domain, we have studied the different propositions that allow the definition of animations. These are mainly programming languages dedicated to animation description, such as Java 3D [7], VRML [5], SVG [6] etc. The conclusion of our study is that such languages are rather low level and take very difficulty into account the dynamic evolution. Having identified interesting facilities proposed by these languages, we defined a higher level animation language (which can be mapped to the above mentioned propositions).

Taken an architecture-centric approach, π-ADL of software architecture descriptions are refined up to animation generation. The refinement is based on the use of animation patterns that have been defined.

The paper will detail the two languages (π-ADL and the animation language) as well as the animation patterns.

The genericity of our proposition relies on several aspects. First, in the architecture description domain, the concepts identified for animation are generic (although illustrated for a specific ADL). Secondly, by proposing a higher level animation language, we remain independent of particular animation implementation: the technical solution can be chosen according to specific needs and changed according to market evolution. Moreover, the refinement process can be parameterised, and thus allows the generation of animations that better suit the user preferences (in terms of detail of information presented, etc.).

References

[1]. Garlan D., Monroe R., Wile D. *Acme : an Architecture Description Interchange Language*. Proceedings of CASCON'97, pp. 169-183. Toronto, November 1997.

[2]. Medvidovic N., Taylor R. *A Classification and Comparison Framework for Architecture Description Languages*. Technical Report UCI-ICS-97-02, Department of Information and Computer Science, University of California. Irvine, February 1997.

[3]. Monroe R. *Capturing Software Architecture Design Expertise With Armani*. Tech. Report CMU-CS-98-163.

[4]. Oquendo F., Alloui I., Cimpan S., Verjus H., *The ArchWare ADL: Definition of the Abstract Syntax and Formal Semantics*, ArchWare European RTD Project IST-2001-32360, Deliverable D1.1b, December 2002.

[5]. *The Virtual Reality Modelling Language Specification Version 2.0*, ISO/IEC CD 14772, August 4, 1996.

[6]. *Scalable Vector Graphics (SVG) 1.0 Specification*, W3C Candidate Recommendation 02 November 2000, http://www.w3.org/TR/2000/CR-SVG-20001102

[7]. Sun Microsystems, *The Java 3DTM API Specification*, Version 1.2, April 2000, http://java.sun.com/products/java-media/3D/forDevelopers/J3D_1_2_API/j3dguide/