Chapter 8

Is There A Life After Objects?

Jean Bézivin
University of Nantes, France

We are presently witnessing a rapid paradigm change in software engineering: from objects to models. This chapter discusses some aspects of the emerging domain of model engineering, mainly those related to meta-modelling and uniform representation of models and meta-models. This much recalls the discussions on classes and objects that were taking place in the early eighties. In the last period, the issue of code interoperability has been dealt with such acceptable solutions as CORBA or the associated IDL language. But we are now witnessing the multiplication of non-executable models, as part of the evolving software development practice. In order to cope with this increasing complexity, a general and regular framework has to be defined. This is being achieved, in environments like the OMG, where all the new models are based on a precise meta-model and where all the meta-models are based on a common and unique meta-meta-model called the MOF. The MOF is rapidly gaining practical importance, between UML and XML, in the industrial strategy of several important companies. It is playing the role of a knowledge bus for all kind of models, object models or legacy models, product models or process models, existing models or yet to be defined models. In particular it is helping to provide a smooth transition from objects and components models to the business processes, workflows and service models that are becoming key elements in the area of Web services.

This chapter appears in the book, Optimal Information Modeling Techniques by Kees van Slooten. Copyright © 2002, IRM Press, an imprint of Idea Group Inc.
FROM OBJECTS TO MODELS

Procedural technology had some recognized weaknesses and this led to the move, in the eighties, to object and component technology. While some of the problems have been fixed, other difficulties are still there. One of the additional drawbacks brought by objects was the lost of functional roots in applications. This problem was only very partially addressed by the introduction of the “use case” concept in analysis and design methods. Today many see the return of the associated concept of “service” as an important turn in the information system landscape, while object and component models are more and more viewed as pertaining to the implementation side. The most common expression of these models of service appears in the design of Web services. Among the reasons for this is the fact that the concept of service may help to conciliate much more easily functional and non-functional requirements (e.g. QoS properties). Also services are more easily combined with business processes and workflows.

Other examples of facets that the object models were not able to readily capture were architecture and know-how information. It was necessary to invent the “design pattern” and other concepts to cope with these. Similarly to services or use cases, design patterns are not usually considered as first class objects in current practices. Many other limitations of object organization of software were subsequently noticed, some of them tentatively expressed in aspect-oriented programming.

Most of these views can only be taken into account in a general framework that is based on regular handling of multiple models like the one currently suggested in the Model-Driven Architecture (MDA) initiative (Soley, 2000). The Semantic Web (Berners-Lee, Hendler & Lassila, 2001) and the MDA are probably among the most significant industrial evolutions in computer science at the beginning of this century and there are some points of similarity between them, even if they apply to different contexts. This chapter discusses the main characteristics of the MDA, concretizing the practical transition from objects to models.

The arrival to maturity of object technologies has allowed the idea of model-based software development to find its way to practicality. The OMG (Object Management Group) is no more centering its activities on a unique interoperability bus, but on two different ones: the classical CORBA software bus (for code interoperability) and the emerging MOF (Meta-Object Facility (OMG/MOF, 1997)) knowledge bus (for model interoperability). The consensus on UML (Unified Modelling Language (Kobryn, 1999)) has been instrumental in this transition from code-oriented to model-oriented software production techniques. A key role is now played by the concept of meta-model in new software organizations like the OMG meta-model stack architecture or the MDC OIM (Open Informa-
Related Content

Integrated Security Process Improvement Framework for Systems and Services
[www.igi-global.com/article/integrated-security-process-improvement-framework-for-systems-and-services/104654?camid=4v1a](www.igi-global.com/article/integrated-security-process-improvement-framework-for-systems-and-services/104654?camid=4v1a)

The MFC Cybersecurity Model Extension and Diagnostic toward a Depth Measurement: E-Learning Systems Case Study
[www.igi-global.com/chapter/the-mfc-cybersecurity-model-extension-and-diagnostic-toward-a-depth-measurement/135228?camid=4v1a](www.igi-global.com/chapter/the-mfc-cybersecurity-model-extension-and-diagnostic-toward-a-depth-measurement/135228?camid=4v1a)
Optimized System-Level Design Methods for NoC-Based Many Core Embedded Systems
[www.igi-global.com/chapter/optimized-system-level-design-methods-for-noc-based-many-core-embedded-systems/116108?camid=4v1a](www.igi-global.com/chapter/optimized-system-level-design-methods-for-noc-based-many-core-embedded-systems/116108?camid=4v1a)

A Review on Vision-Based Hand Gesture Recognition and Applications
[www.igi-global.com/chapter/a-review-on-vision-based-hand-gesture-recognition-and-applications/135889?camid=4v1a](www.igi-global.com/chapter/a-review-on-vision-based-hand-gesture-recognition-and-applications/135889?camid=4v1a)