Chapter XVIII

Towards a UML Profile for Building on Top of Running Software

Isabelle Mirbel
Laboratoire I3S, France

Violaine de Rivieres
Amadeus sas, France

ABSTRACT
Currently, fewer and fewer applications are developed from scratch. Therefore, with any development process, it is very important to determine during the analysis and design phases whether there are any applications that must be safeguarded and how this could be accomplished. Legacy applications, as well as Enterprise Resource Planning integration are typical examples of developments that deal with safeguarding. Indeed, safeguarding may be necessary for a specific piece of work involving the integration of new developments with different parts of running applications. To support such a difficult but fundamental task, we recommend a set of extensions through a UML profile. In this proposal, we highlight three aspects of safeguarding which have to be taken into account: the business expertise, the interfaces, and the code itself. We then present how this profile can be used along the different phases of analysis and design; applicable guidelines are provided to support software designers in their daily work.

INTRODUCTION
New applications are constantly built on top of running ones. Legacy applications and Enterprise Resource Planning (ERP) are typical examples of development on top of
Towards a UML Profile for Building on Top of Running Software

BACKGROUND

Many developments now start from running software rather than from scratch. This considerably changes the way analysis and design has to be handled: different aspects of the running software must be taken into consideration for the future development. In addition to the code itself, functional domain expertise (i.e., expertise taken from the functionalities, data, and screen shots) may also be of interest. Interfaces describing relationships that the running software has with other systems (software, databases, etc.) should also be taken into consideration early on in the software development process. But such a situation is rarely handled in the current analysis and design approaches (Spit, 1995).

Problems related to building on top of running software have been studied before, but only from the implementation point of view (Fowler, 1999; Beck, 1997; Opdyke, 1992). We believe that any issues should already be taken into consideration during the analysis phase. Undeniably, in addition to the code itself, the expertise regarding the functional domain and the interfaces (describing any relationships that the running
Related Content

Sharing Ontologies and Rules Using Model Transformations
[www.irma-international.org/chapter/sharing-ontologies-rules-using-model/35871/](www.irma-international.org/chapter/sharing-ontologies-rules-using-model/35871/)

Modeling of Web Services using Reaction Rules
[www.irma-international.org/chapter/modeling-web-services-using-reaction/35869/](www.irma-international.org/chapter/modeling-web-services-using-reaction/35869/)

XML Stream Query Processing: Current Technologies and Open Challenges
Mingzhu Wei, Ming Li, Elke A. Rundensteiner, Murali Mani and Hong Su (2009). *Open and Novel Issues in XML Database Applications: Future Directions and Advanced Technologies* (pp. 89-107).
[www.irma-international.org/chapter/xml-stream-query-processing/27778/](www.irma-international.org/chapter/xml-stream-query-processing/27778/)

Using Device Detection Techniques in M-Learning Scenarios
[www.irma-international.org/chapter/using-device-detection-techniques-learning/73176/](www.irma-international.org/chapter/using-device-detection-techniques-learning/73176/)

Developing Software Testing Ontology in UML for a Software Growth Environment of Web-Based Applications
[www.irma-international.org/chapter/developing-software-testing-ontology-uml/29616/](www.irma-international.org/chapter/developing-software-testing-ontology-uml/29616/)