

Model-Driven Architecture in Practice

A Software Production Environment Based on Conceptual Modeling

Bearbeitet von
Óscar Pastor, Juan Carlos Molina

1. Auflage 2007. Buch. xvi, 302 S. Hardcover
ISBN 978 3 540 71867 3
Format (B x L): 15,5 x 23,5 cm
Gewicht: 643 g

[Weitere Fachgebiete > EDV, Informatik > Software Engineering > Modellierung, UML, SysML](#)

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei


DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Foreword

Not another OO methodology book, I hear you say. But you are wrong. This is the first of a new trend that makes MDD (model-driven development) a reality. Written collaboratively by the inventor of OO-Method and OASIS (Pastor) and a leading industry spokesman who has put Pastor's ideas into practice (Molina), this offers both a rigorous academic treatment of conceptual modelling as a precursor to full code generation and a practitioner's insights into realizing this dream.

The book commences with philosophy, argues that existing modelling languages are not formal enough to create anything more than code skeletons and then describes in detail the necessary semantics that a conceptual modelling language needs for MDD. This means that, for an industry development team, the abstraction level is now raised. No longer do they need to worry about technicalities (and "features") of programming languages but can devote themselves to the less mechanistic and more creative parts of software engineering: analysis and design.

There are a number of models (strictly viewpoints) presented: a structural model, a behavioural model, a functional model and a presentation model. These emphasize different aspects of the (single) underlying reality and the conceptual schema that characterizes it in the software modelling domain. It is stressed how these must be a single whole, integrated in such a way that the final transformation to code is accomplished as a single coherent action.

Of critical importance from an industry perspective is the Presentation Model. Many authors would assume that this should not be part of a theoretical development or a conceptual modelling language – tools do that, don't they? However, the success or failure of any particular software application depends on the user's view of it. If it is difficult or clumsy to use, then it won't be used. Thus these sections on presentation modelling will be critical to any tool's success, including the OlivaNova tool described here in the last section of the book.

Of course, there are many other research and industry efforts to create high quality MDD tools. Some of these are discussed and, although the authors are clearly promoting their own way, the analysis of the alternatives is rational and unemotional.

Nor is this just a greenfield discussion – legacy systems are also discussed in the MDD context.

The book also has a running example of a car rental system. Not only is this easy to comprehend but, as it happens, I have seen the generated system running successfully.

So beware. If you are a UML addict, don't be blinded by the bits of OASIS and OO-Method that look somewhat like UML. Read the details that make this approach really viable for full code generation. See what can be done by a real collaboration between academe and industry. Enjoy.

Geneva,
February 2007

Brian Henderson-Sellers