How this Book Is Organized

Programming technologies have improved continuously during the last decades but, from an Information Systems perspective, some well-known problems associated with the design and implementation of an Information Systems persist: Object-Oriented Methods, Formal Specification Languages, Component-Based Software Production, Aspect-Oriented Approaches. This is only a very short list of technologies proposed to solve a very old and, at the same time, very well-known problem: how to produce software of quality. Programming has been the key task during the last 40 years, and the results have not been successful yet. This book will explore the need of facing a sound software production process from a different perspective: conceptual model-based software production. There are several ways to refer to that strategy. There are people dealing with the non-programming perspective where, by non-programming, we mean mainly modelling. Rather than Extreme Programming, the issue is that an Extreme Non-Programming (Extreme Modelling-Oriented) approach should be taken.

Other people focus on Conceptual Schema-Centred Software Production, based on the assumption that, to develop an Information System (IS), it is necessary and sufficient to define its Conceptual Schema. This is presented in (Olivé 2005) as a grand challenge for Information Systems Research. This book is oriented to face this challenge, providing concrete solutions. In particular, we will show how to achieve the objective of generating code from a higher-level system specification, normally represented as an Object-Oriented Conceptual Schema. Nowadays, the interest in MDA has provided a new impetus for all these strategies. New methods propose different types of model transformations that cover all the different steps of a sound software production process from an Information Systems Engineering point of view. This must include Organizational Modelling, Requirements Engineering, Conceptual Modelling and Model-Based Code Generation techniques. In this context, it seems that the time of Model-Transformation Technologies is finally here.

Under the push of this technological wave, and taking advantage of our years of experience working on Model-Driven Development, we will defend the main idea that, to have a software product of quality, the key skill is modelling; the issue is

that "the model is the code" (rather than "the code being the model"). Considering this hypothesis, a sound Software Production Process should provide a precise set of models (representing the different levels of abstraction of a system domain description), together with the corresponding transformations from a higher level of abstraction to the subsequent abstraction level. For instance, a Requirements Model should be properly transformed into its associated Conceptual Schema, and this Conceptual Schema should be converted into the corresponding Software Representation (final program).

Assuming that, behind any programmer decision, there is always a concept, the problem to be properly faced by any Model-Transformation Technology is that of accurately identifying those concepts, together with their associated software representations. A precise definition of the set of mappings between conceptual primitives or conceptual patterns and their corresponding software representations provides a solid basis for building Conceptual Model Compilers.

This is what we wish to do in this book. Our precise objective is to show how an MDA-based Software Production Environment based on Conceptual Modelling can be put into practice. To do this, three main goals need to be fulfilled. First, the main concepts involved in such a method must be properly introduced. Second, how to construct an adequate Conceptual Model has to be explained in detail. Third, the process of transforming the source Conceptual Model into its corresponding Software Product must be explained. In accordance with these goals, the structure of this book is divided into three parts:

- The OO-Method and Software Production from Models
- Conceptual Modelling: About the Problem Space
- Conceptual Model Compilation: from the Problem Space to the Solution Space

Now it is time for the reader to explore how to put all these ideas into practice, making Conceptual Model-Driven Development an affordable dream.