1. Record Nr. UNINA9910829108803321

| | |
|---|---|
| Autore | Kelly Steven |
| Titolo | Domain-specific modeling : enabling full code generation / / Steven Kelly, Juha-Pekka Tolvanen |
| Pubbl/distr/stampa | Hoboken, N.J., : Wiley-Interscience, : IEEE Computer Society, c2008 |
| ISBN | 1-281-21727-1<br>9786611217273<br>0-470-24926-9<br>0-470-24925-0 |
| Edizione | [1st ed.] |
| Descrizione fisica | 1 online resource (445 p.) |
| Altri autori (Persone) | TolvanenJuha-Pekka |
| Disciplina | 005.1 |
| Soggetti | Programming languages (Electronic computers)<br>Computer software - Development |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Description based upon print version of record. |
| Nota di bibliografia | Includes bibliographical references (p. 415-421) and index. |
| Nota di contenuto | Foreword -- Preface -- PART 1: BACKGROUND AND MOTIVATION -- 1. Introduction -- 1.1 Seeking the better level of abstraction -- 1.2 Code-driven and model-driven development -- 1.3 An example: modeling with a general-purpose language andwith a domain-specific language -- 1.4 What is DSM? -- 1.5 When to use DSM? -- 1.6 Summary -- 2. Business value -- 2.1 Productivity -- 2.2 Quality -- 2.3 Leverage expertise -- 2.4 The economics of DSM -- 2.5 Summary -- PART 2: FUNDAMENTALS -- 3. DSM defined -- 3.1 DSM characteristics -- 3.2 Implications of DSM for users -- 3.3 Difference to other modeling approaches -- 3.4 Tooling for DSM -- 3.5 Summary -- 4. Architecture of DSM -- 4.1 Introduction -- 4.2 Language -- 4.3 Models -- 4.4 Code generator -- 4.5 Domain framework and target environment -- 4.6 DSM organization and process -- 4.7 Summary -- PART 3: DSM EXAMPLES -- 5. IP telephony and call processing -- 5.1 Introduction and objectives -- 5.2 Development process -- 5.3 Language for modeling call processing services -- 5.4 Modeling IP telephony service -- 5.5 Generator for XML -- 5.6 Framework support -- 5.7 Main results -- 5.8 Summary -- 6. Insurance products -- 6.1 Introduction and objectives -- 6.2 Development process -- 6.3 Language for modeling |

insurances -- 6.4 Modeling insurance products -- 6.5 Generator for Java -- 6.6 Framework support -- 6.7 Main results -- 6.8 Summary -- 7. Home Automation -- 7.1 Introduction and objectives -- 7.2 Development process -- 7.3 Home automation modeling language -- 7.4 Home automation modeling language in use -- 7.5 Generator -- 7.6 Main results -- 7.7 Summary -- 8. Mobile phone applications using Python framework -- 8.1 Introduction and objectives -- 8.2 Development process -- 8.3 Language for application modeling -- 8.4 Modeling phone applications -- 8.5 Generator for Python -- 8.6 Framework support -- 8.7 Main results -- 8.8 Extending the solution to native S60 C++ -- 8.9 Summary -- 9. Digital Wristwatch -- 9.1 Introduction and Objectives.
9.2 Development Process -- 9.3 Modeling Language -- 9.4 Models -- 9.5 Code Generation for Watch Models -- 9.6 The Domain Framework -- 9.7 Main Results -- 9.8 Summary -- PART 4: CREATING DSM SOLUTIONS -- 10 DSM language definition -- 10.1 Introduction and objectives -- 10.2 Identifying and defining modeling concepts -- 10.3 Formalizing languages with metamodeling -- 10.4 Defining language rules -- 10.5 Integrating multiple languages -- 10.6 Notation for the language -- 10.7 Testing the languages -- 10.8 Maintaining the languages -- 10.9 Summary -- 11. Generator definition -- 11.1 "Here's one I made earlier" -- 11.2 Types of generator facilities -- 11.3 Generator output patterns -- 11.4 Generator structure -- 11.5 Process -- 11.6 Summary -- 12. Domain Framework -- 12.1 Removing duplication from generated code -- 12.2 Hiding platform details -- 12.3 Providing an interface for the generator -- 12.4 Summary -- 13. DSM definition process -- 13.1 Choosing among possible candidate domains -- 13.2 Organizing for DSM -- 13.3 Proof of concept -- 13.4 Defining the DSM solution -- 13.5 Pilot project -- 13.6 DSM deployment -- 13.7 DSM as a continuous process in the real world -- 13.8 Summary -- 14. Tools for DSM -- 14.1 Different approaches to building tool support -- 14.2 A Brief History of Tools -- 14.3 What is needed in a DSM environment -- 14.4 Current tools -- 14.5 Summary -- 15. DSM in use -- 15.1 Model reuse -- 15.2 Model sharing and splitting -- 15.3 Model versioning -- 15.4 Summary -- 16. Conclusion -- 16.1 No sweat shops -- But no Fritz Lang's Metropolis either -- 16.2 The onward march of DSM -- Appendix A: Metamodeling Language. -- References -- Index.

Sommario/riassunto

"[The authors] are pioneers. . . . Few in our industry have their breadth of knowledge and experience." / -From the Foreword by Dave Thomas, Bedarra Labs Domain-Specific Modeling (DSM) is the latest approach to software development, promising to greatly increase the speed and ease of software creation. Early adopters of DSM have been enjoying productivity increases of 500-1000% in production for over a decade. This book introduces DSM and offers examples from various fields to illustrate to experienced developers how DSM can improve software development in their teams.Two authorities in the field explain what DSM is, why it works,and how to successfully create and use a DSM solution to improve productivity and quality. Divided into four parts, the book covers:background and motivation; fundamentals; in-depth examples; and creating DSM solutions. There is an emphasis throughout the book on practical guidelines for implementing DSM, including how to identify the necessary language constructs, how to generate full code from models, and how to provide tool support for a new DSMlanguage. The example cases described in the book are available the book's Website, www.dsmbook.com, along with, an evaluation copy of the MetaEdit+ tool (for Windows, Mac OS X, and Linux), which allows readers to examine and try out the modeling

languages and code generators.Domain-Specific Modeling is an essential reference for lead developers, software engineers, architects, methodologists, and technical managers who want to learn how to create a DSM solution and successfully put it into practice.