

FOREWORD

I have been an enthusiastic follower of Juha-Pekka Tolvanen and Steven Kelly's work since meeting them in the 1990s at ECOOP and OOPSLA conferences. When people mention the talented minds of Finland, my first association is not Nokia or Linux, but MetaCase.

I have spent my career searching for ways to empower application and product developers who have domain knowledge to simply and quickly express their knowledge in a form that can be readily consumed by machines. In almost every case, this has led to a little language expressed in text, diagrams or a framework in a friendly OO language such as Smalltalk or Ruby. Today we call these little languages Domain Specific Languages.

Domain Specific Language engineering is increasingly recognized as an important productivity technique which increases businesses' agility in the design, development and configuration of their products. DSLs provide a means for narrowing the communication gap between users and developers. They allow requirements and specifications to be made more tangible to users and developers. Finally, they document critical portions of the knowledge associated with an application or product thereby reducing the life cycle costs in evolving that application.

Juha-Pekka Tolvanen and Steven Kelly are pioneers in the world of DSLs. These experts have worked for over a decade in the design of DSLs and the implementation of commercial DSL tooling at MetaCase. Their popular tutorials and workshops have been featured at major conferences all over the world. Juha-Pekka and Steven have worked closely with customers to implement DSLs specific to their needs. Few in our industry have their breadth of knowledge and experience.

When I first encountered their work I assumed from the name MetaCase and the demonstrations that it was a very clever constraint drawing framework which could be

used to build visual modeling tools. Their frameworks and tools made it easy to express a custom visual notation allowing one to have a full-blown visual modeling tool in weeks rather than years. Anyone with experience using modern graphical modeling frameworks such as Eclipse GMF will be very impressed with how quickly one can define a new visual language.

I soon learned that their real interests were not confined to visual frameworks, but rather they too had a passion for domain specific languages. They partnered with their clients to help them model, design and implement DSLs for their business needs. In doing so they developed the process, practices, tooling and most importantly pragmatics for the industrial use of DSLs.

This book presents practical design and development lessons covering topics including domain modeling, language definition, code generation and DSL tooling. They thoroughly discuss the business and technical benefits of using a DSL. While being proponents of the approach they provide sound arguments for when it is appropriate to consider using a DSL. Importantly, they explore issues associated with the evolution of software using domain specific languages.

The case studies in telephony, insurance, home automation and mobile applications clearly illustrate the use of DSLs for different domains and are based on actual client experiences. They provide the reader with the benefit of a real world perspective on DSL design and implementation. Students and Educators will appreciate the Digital Watch which is a complete pedagogical example used in their popular tutorials.

I have had the pleasure of observing the authors during their journey from research to practice. Far too often the principals in small technology companies are too busy doing to take time to share their unique experiences. We are very fortunate that Juha-Pekka and Steven have made the effort to produce this practical book based on their experiences. I have attended their tutorials and read drafts of the book, each time learning something new. This book is a must read for anyone who wants to understand the appropriate use, benefits and practices of DSL engineering.

DAVE THOMAS