

## Foreword

It is with great pleasure that we gather in this volume the papers given at the 2002 VDM Workshop, part of the Formal Methods Europe Symposium 2002, itself a part of the 2002 Federated Logic Conference. This is the third VDM Workshop in recent years and is in some ways the most wide-ranging.

VDM, the Vienna Development Method, is one of the longest-established formal methods. It utilises a modelling, or specification, language based on abstract data structures with functionality expressed through explicit referentially transparent functions, imperative algorithms or by implicit pre-/post-conditions. The modelling language has an ISO-Standard syntax and semantics, as well as an underlying proof and refinement theory. Tools support the development of VDM models in modular and object-oriented styles, as well as testing, validation and proof.

Much of the work on VDM in recent years has been driven by its use on the “factory floor” of systems development – the VDM community has a strong pragmatic streak. However, it would be a mistake to treat the method as “just” the standard language and popular tools. The VDM notations and principles have provided a flexible, abstract basis for research that steps beyond the constraints of software engineering practice into areas such as language design, proof theory, data structures and algorithmics. The papers in this third workshop reflect this breadth of application.

José Oliveira, our invited contributor, presents an application of VDM in the experimental spirit. The relational model of functionality provides a basis for a suggested taxonomy of algorithms, revealing structure and suggesting “missing” information that could lead to novel algorithms in the future.

The importance of practical application is brought out by Keijiro Araki and Han-Myung Chang in their frank assessment of the “pre-myth” state of formal methods in Japan. They stress the importance of positive and demonstrable successful applications, as well as training and tool support, in the drive to improve software development quality in a cost effective manner. Natsuki Terada reports on just such a practical study demonstrating advanced tool support for proof-based analysis of models using the new HOL-based proof system developed in the PROSPER project.

Good structuring mechanisms are at least as important in formal modelling as they are in programming for coping with the scale and complexity of substantial applications. Nazareno Aguirre, Juan Bicarregui, Theo Dimitrakos and Tom Maibaum draw on experience with both VDM and B to consider the extension of module/object mechanisms in ways that have minimum impact on the semantics of the underlying flat language.

The potential of formal models in test generation and application is still being realised. Fundamental issues such as test selection, ordering and reification continue to be addressed in the practical context of tool development. Olivier Maury, Yves Ledru, Pierre Bontron and Lydie du Bousquet report work on a tool to support test engineers in the application of test selection policies. Alexander Koptelov, Victor Kuliainin and Alexander Petrenko tackle the issue developing partition tests over VDM++ models containing program-like components allied to pre-/post-condition specifications.

In their tutorial on VDM++ , John Fitzgerald, Peter Gorm Larsen, Paul Mukherjee and Nico Plat present key aspects of VDM++ including its approach to OO and concurrency and its features for analysis and synthesis of models. They also give an overview of the available tool support and summarise some industrial experiences of its use.

This small collection of contributions gives a “snapshot” view of the breadth of work using VDM today. We look forward to future work, and to future workshops.

J C Bicarregui  
J S Fitzgerald  
P G Larsen

Copenhagen, July 2002