



The First Workshop on Behavioural Modelling  
in Model-Driven Architecture (BM-MDA)

# The First Workshop on Behavioural Modelling in Model-Driven Architecture (BM-MDA)

Organizers:

Mehmet Aksit, TU Twente, the Netherlands

Ekkart Kindler, Technical University of Denmark

Ella Roubtsova, Open University of the Netherlands

Ashley McNeile, Metamaxim Ltd, UK



The First Workshop on Behavioural Modelling  
in Model-Driven Architecture (BM-MDA)

**What are the major problems  
with behavioural modelling in  
MDA?**

- Comparing behaviour (verification, validation) Hajo Eichler
- Defining high-level description of languages, SOA, expressing behaviour, extracting behaviour (Selo Slistyo)
- To what extent can models be used in real development, SPL, integration of views (Tim Trew)
- Supporting MDA in various ways, behaviour modelling notations that will be used by practitioners, graphical or textual?, how to address many/wide domains (Audris Kalnins)
- Telelogic/Rhapsody, workflow, integration of model and implementation views
- Model at different abstraction levels, refinement and verification, mobile applications, (Luis Pires)
- ASML, model driven engineering in industry, DSL
- Integration of business systems like SAP, integration of behavioural aspects, interfaces
- Run time verification, possibility of a common notation for verification, composing behaviour and data views
- User of MDA tool, re-use, creating trivial functionality easily
- Generating model transformations, patterns for transformations
- Model refinement, guaranteeing correctness by transformation, using patterns of behaviour, combining patterns to create executable code
- Comparing behaviour to prove that the model is preserved by transformation
- Combine formal methods and MDA/MDE, defining semantics, dependable systems
- Combining light weight modelling with formal methods and programming languages, dependable adaptive systems, role of models in adaptive systems
- What is the purpose of behaviour modelling? Is it possible to make behaviour modelling easy or is it bound to be as complex as a programming language. Must we accept this complexity?
- Software services provider, tool development. Spread the word, Clear concepts and models, separation of concerns in modelling, achieving quality, how to convince the public
- Select tools and set practices in software design. Embedded: Possible – Rhapsody. Business: Enterprise Architect – only uses structural code generation. Can you model entire behaviour. Recovery of behavioural design/model from legacy applications (structural is easier)
- Is complete code generation really wanted? Is it easier than writing code? Aspect Oriented Modelling.