Editorial: Model-based Requirements Engineering

Tim Menzies
Lane Department of Computer Science,
University of West Virginia,
PO Box 6109, Morgantown,
WV, 26506-6109, USA;
http://tim.menzies.com;
tim@menzies.com

March 30, 2003

Aristotle once said that ”it is the mark of an educated mind to be able to entertain a thought without accepting it”. This is the motto of the model-based requirements engineering (MBRE). MBRE analysts know that defining requirements means herding a community of stakeholders through a space of options towards some group consensus. Left to themselves, stakeholders may argue forever about different issues. The MBRE premise is that a model of the current version(s) of the RE model should be treated like a *laboratory* where analysts can experiment with different options. Such experimentation can:

- Inform the arguments with new information gleamed from the models;
- Comment and assess the current RE model;
- Shorten the time taken for those arguments;
- Reach a resolution that better satisfies feuding stakeholders;
- Ensure better decisions

Two objections to MBRE might be:

- “MBRE is impractically expensive since it takes too long to build the RE model.”
- “Even if early life cycle models exists, then no effective conclusions can be drawn from them since they are incomplete and full of contradictions and overlaps.”

The papers in this special issue form an eloquent and comprehensive reply to these objections. For example:

- In *Merging Individual Conceptual Models of Requirements*, Richards shows that the natural language seen in commonly-used RE methods (i.e. UML use cases) are sufficiently structured to support contradiction and subsumption detection.

This finding would be enough to merit inclusion in this special issue. However, Richards goes further and augments her tools with semi-automatic methods that help stakeholders resolve their feuds.

A surprising feature of these papers is how much precision and automatic support can be applied to models very early in the RE process. The SPIN model checker is an automatic formal verification tool developed to verify concurrent protocols in software. Surprisingly, the same tool can be applied to early life cycle RE models:

- In *Lightweight models for interpreting informal specifications*, Fuhrman shows how RE models can be refined using SPIN as an *ambiguity detector*. In his preferred iterative RE process, SPIN is not used to *validate* a model. Rather, it us used to generate a *commentary* on analyst's current interpretation of the specification.

Another application of SPIN can be found in:

- *Arcade: Early Dynamic Property Evaluation of Requirements Using Partitioned Software Architecture Models*- In this paper, Barber *et.al* augment standard RE with a suite of model-based tools such as model checkers like SPIN, discrete event simulators, and probabilistic graph model algorithms. Their suite can comment on issues such as correctness, performance, and reliability of early life cycle models.

MBRE tools must span the spectrum of model formality from very lightweight to very formal (such very formal models can appear in the requirements of ultra-safety critical applications). Such a spectrum can be seen in this issue. For example, if RE models are not machine readable, they can still be read by humans:

- In *An Economic Approach for Improving Requirements Negotiation Models with Inspection*, Halling *et.al* explore the merits of humans inspecting requirements models using a variant of *formal inspection methods*.These methods have been used for decades to read and assess source code. But Halling *et.al.* show that inspections are an economic validation technique for models built to support negotiation between different stakeholders.

Moving along the spectrum we see methods for handling natural language (such as the Richard's paper described above) and modelling frameworks where users can hastily jot down their concepts. For example:

- In *Quantitative Risk-Based Requirements Reasoning*, Feather & Cornford describing a visual modelling tool used with groups of experts in real-time as they meet to debate requirements options. Their *risk-balancing* approach interactively and visually tracks the cost-benefit implications of various project options.

Moving even further along the spectrum towards fully formal models we see the systems of Fuhrman and Barber *et.al.* described above. Finally, the most formal paper of this issue discusses formal *retrenchments*:

- In *Retrenching Partial Requirements into System Definitions: A Simple Feature Interaction Case Study* , Banach and Poppleton offers a framework for the stepwise, layered construction of a requirements specification, accounting for both beneficial and harmful interactions. For harmful, or interfering, interactions, the framework allows design revisions to resolve the interaction in the layered construction process.

This special issue is the result of much work by many people. Some of the material in this issue are elaborations of papers first presented at the International Workshop on MBRE (San Diego, USA December, 2001) co-chaired by Jim Kipers, Tim Kurtz and myself. Thanks should also go to the REJ co-editor Peri Loucopoulos and the reviewers of this special issue who all completed their review assignments exactly on time!

Tim Menzies
Morgantown, West Virginia
September 2002.