Algebraic Specification in computer-aided multi-paradigm modelling

Ernesto Posse

April 20, 2006

Computer-aided multi-paradigm modelling (CAMPAM, [5]) promises to help tackling complexity in modelling, design and analysis of complex systems, whether they are physical systems, software systems or any other kind. It encompasses the study of model-abstraction, multi-formalism modelling and meta-modelling.

There are many issues in CAMPAM that need to be addressed before it can deliver its promises. Some issues concern the foundations of multi-paradigm modelling itself, while others concern the "computer-aid" part of CAMPAM. Here, I identify some of the issues which I consider to be amongst the most relevant, and I propose the use of Algebraic Specification [8] as a tool to address these issues.

Foundations of multi-paradigm modelling

Before developing tools to support multi-paradigm modelling, and even before discussing future research directions, it is necessary to answer the following questions:

1. What are the problem(s) we are trying to solve?
2. What is "multi-paradigm modelling?"
3. Does multi-paradigm modelling solve, or help solve the problem(s)?
4. What is a model?
5. What is a meta-model?
6. What is model-abstraction?
7. What is a multi-formalism model?

We can of course, start with some informal answers to these questions, but sooner or later we will find ourselves "chasing rainbows" unless we have precise definitions of the concepts of model, meta-model, model abstraction and multi-formalism models.

There are, of course, many possible answers to these questions, and many ways to answer them. I would like to suggest an approach to tackle this problem.

In order to be precise, and yet general enough, one approach is to use the language of Mathematics to provide the definitions needed. By doing so, we would be able to use all the mathematical tools at our disposal, not only to answer these questions, but to reason about these concepts beyond the basics.

One of the tools from Mathematics that could provide a fruitful framework is Algebra.

Algebra could help not only in defining concepts such as model and meta-model, but also in providing general *and* useful modelling frameworks, and even serve as the basis for advanced tool support.

Algebra can be useful in many ways. Universal Algebra and Algebraic Specification seem particularly promising. In particular, Algebraic Specification can be seen as a tool to describe models and meta-models.

Algebraic Specification is a methodology where a set of mathematical objects are defined by providing their structure, usually in terms of sets and operations on these sets, and a set of axioms which must be satisfied by the objects being defined. Any typical mathematical definition can be seen as an example. In computing the typical application is the definition of abstract data types.

Algebraic Specification is also closely related to Type-Theory: the later can provide mechanisms to verify that the signatures of an algebraic specification is correct. Furthermore, it has close ties with Logic, since axioms are a fundamental part of specifications.

How can we apply Algebraic Specification to multi-paradigm modelling? One approach is to consider meta-models to be *signatures* or *specifications* (in the sense of [8]) and models to be *algebras* satisfying those specifications. In this view, many definitions follow naturally: the meaning of a meta-model could be considered to be the set of algebras satisfying it. Structured algebraic specifications also provide a means to combine simpler specifications into composite ones, thus suggesting a way to combine meta-models meaningfully, a central issue in multi-formalism modelling. Many Algebraic Specification concepts can be interpreted in the multi-paradigm modelling context. For example, model-abstraction could be defined in terms of the traditional notion of homomorphism from Universal Algebra. The algebraic notion of *terms* with variables could be seen models with "placeholders" that could be replaced by other (sub)models. Parametrized algebraic specifications could be seen as parametrized meta-models.

A common approach to meta-modelling is to represent models and meta-models as graphs. Some interesting questions arise. Is the purely graphical approach better than the algebraic specification approach, or vice-versa? The algebraic specification approach seems more general[1], yet, the graphical approach is very attractive. It might be desirable to have the best of both worlds. Can the two approaches be integrated? This suggests a possible candidate for the definition of model and meta-model: a combination of graphs and algebraic specifications. This is a question that deserves to be explored.

Further relevant issues in CAMPAM where Algebraic Specification may help include: model and system composition, refinement, and verification.

On the issues of composition Algebraic Specification can help in several ways. There are many important questions that arise because of composition: can components be composed to form a more complex system? how can components be composed? can the meaning of a composite be given in terms of its components? can we verify a property for the whole if we can verify the property for its parts? In order to answer these, and many others, we need to define composition itself. Algebraic Specification can be useful in at least two ways: first, by means of addressing these questions in a context of structured Algebraic Specifications, where models are algebras and composition is given by composition of signatures and specifications; second, by providing a means to define the syntax and semantics of *composition languages*.

A second issue is that of refinement, closely related to abstraction. Here there are multiple relevant questions, such as: given an abstract model, how can we obtain a *refinement* of it, closer to an implementation? Can we obtain a minimal, canonical or optimal refinement of a given model? Given two models how do we tell whether one is a refinement of the other? Some notions from Universal Algebra can be applied to address these questions.

Finally the issue of verification. The fundamental problem is this: given a model and a property, does the model satisfy the property? Two fundamental approaches to this problem fit naturally in the context of Algebraic Specification: the axiomatic, or proof-theoretic approach, and the model-checking approach. In the first approach verifying a property consists of deducing it from the axioms of a specification. This is particularly useful for properties that should be satisfied by all models in the class defined by the specification. In the second, the property is checked against the model itself. This is appropriate for model-specific properties.

Computer-support in CAMPAM

Answering some of the foundational questions suggests a number of useful tools to support the process of multi-paradigm modelling. One of the advantages of using Algebraic Specification is that there are already tools that support this approach, for example, the Common Algebraic Specification Language (CASL [1]) based on ASL [7], and other languages such as CLEAR [2], OBJ2 [4], ACT ONE [3], ACT

[1] Certainly a graph can be defined algebraically.

TWO [6], etc.

Nevertheless, these languages and tools do not support directly the view of "models as graphs" and thus, some new tools are desirable. This includes tools for editing such models, and in particular tools that support structured or composite specifications; tools for model exchange, model transformation, and specification "management"; and tools for analysis, namely for *refinement* and abstraction checking, and for verification.

Concluding remarks

I believe that Algebraic Specification can help CAMPAM as described above and possibly in other ways, but fundamentally the main benefit to be gained is to provide structure and organization to CAMPAM.

References

[1] Michel Bidoit and Peter D. Mosses. CASL *User Manual.* LNCS 2900 (IFIP Series). Springer, 2004. With chapters by T. Mossakowski, D. Sannella, and A. Tarlecki.

[2] R. M. Burstall and J. A. Goguen. The semantics of CLEAR, a specification language. In D. Bjorner, editor, *Advanced Course on Abstract Software Specifications*, volume 86 of *Lecture Notes in Computer Science*, pages 292–332. Springer, 1980.

[3] H. Ehrig, W. Fey, and H. Hansen. An algebraic specification language with two levels of semantics. Bericht 83-03, Technische Univ. Berlin, 1983.

[4] K. Futatsugi, J. A. Goguen, Jouannaud J.-P., and J. Meseguer. Principles of OBJ2. In *Proc. of POPL*, pages 52–66, 1985.

[5] Pieter J. Mosterman and Hans Vangheluwe. Computer Automated Multi-Paradigm Modeling. *ACM Transactions on Modeling and Computer Simulation*, 12(4), 2002.

[6] H. Weber and H. Ehrig. Programming in the large with algebraic module specifications. In *Proc. IFIP 10th World Computer Congress*, pages 675–684. North-Holland, 1986.

[7] Martin Wirsing. Structured Algebraic Specifications: a kernel language. *Theoretical Computer Science*, (43):123–250, 1986.

[8] Martin Wirsing. Algebraic Specification. In Jan Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, Formal Models and Semantics, chapter 13, pages 675–788. Elsevier, MIT Press, 1994.