



Preface

Volume 44, Issue 3

Wolfram Kahl, David L. Parnas and Gunther Schmidt

Abstract

RelMiS 2001

This issue contains the *Proceedings* of the First International Workshop on **Relational Methods in Software** (RelMiS 2001). The Workshop was held in Genova, Italy, on April 7 and 8, 2001, as a satellite event to ETAPS 2001.

The rôle of the calculus of relations in algebra and logic is well understood and appreciated; relational methods should be part of the “toolbox” of everyone who uses mathematics, and relational methods can also be of great value to everyone who develops or specifies software. For example, much of the work on special logics or “laws” for programs can be simplified by applying relational algebra. Further, every relation on a state space can be used as a specification or description of a program. This is an advantage over some predicate-based formalisms in which one can write “specifications” that are impossible to satisfy.

During the 1980’s, the equational theory of relation algebras was applied to program semantics and program development. On the other hand, a topic such as relational databases was an essentially new development, but one that did not take full advantage of earlier mathematical work on relations. It has contributed new concepts.

Computer science, as a new application field for relational methods, has both drawn from and contributed to previous work in logics and mathematics. Further integration of the mathematical work in relational methods with research in computer science can only benefit both.

Although a series of international “RelMiCS” workshops on “Relational Methods in Computer Science” already has been established, and, more recently, the joint European COST Action 274 TARSKI (Theory and Application of Relational Methods as Knowledge Instruments) has gained considerable interest, we felt that more exposure of relational methods to software practitioners and computer scientists with interests in software development was needed. Consequently, we decided to initiate a satellite event for ETAPS 2001, and are grateful to the ETAPS organisers for including RelMiS among their satellite events.

RelMiS 2001 consisted of two distinct parts:

- a tutorial introduction to relational methods for computer scientists and software developers, and
- a workshop to discuss new results and possible future work.

For the tutorial part, four lectures were held:

Gunther Schmidt:
Basics of Relational Methods

1. This tutorial presented the foundations of relational methods, stressing the component-free compositional approach which allows more compact formalisations and concise, easily verified, proofs.

David L. Parnas:

The Tabular Method for Relational Documentation

1. The basic mathematical laws of relations can be presented and verified using compact abstract notation. However, descriptions of specific relations, such as those that arise in describing software, must be done in terms of a state representation. When conventional notation is used, the expressions can be complex and hard to use. This tutorial presented and explained tabular notation that has been found useful in practical software development. It also described a set of prototype tools that make it easier to apply relational methods in software development.

Wolfram Kahl:

Refinement and Development of Programs from Relational Specifications

1. Starting from relational specifications, refinement steps may be used to obtain first more detailed specifications, and then programs. The foundations of relational refinement concepts were discussed and the “demonic calculus of relations” and other tools for calculational program development by relational refinement were presented.

Rudolf Berghammer (Christian-Albrechts-Universität zu Kiel):

Prototyping and Programming with Relations

1. This tutorial showed an approach to programming directly on the level of binary relations, and with strong visualisation support. This was then used for solving combinatorial problems and for prototyping complex algorithms on a high level of abstraction.

For the open workshop, we called for contributions in a field centering around the following topics:

Relational Specifications and Modelling

1. Methods and tools, tabular methods, abstract data types

Relational Software Design and Development Techniques

1. Relational refinement, heuristic approaches for derivation, correctness considerations, dynamic programming, greedy algorithms, catamorphisms, paramorphisms, hylomorphisms and related topics

Programming with Relations

1. Prototyping, testing, fault tolerance, information systems, information coding

Handling of Large Relations

1. Problems of scale, innovative representations, distributed implementation, implementing relational algebra with mixed representation of relations

This special issue begins with slides from the second tutorial and notes for the last two tutorials. The remainder is made up of the six accepted contributions presented on the second day of the ReMiS workshop, which saw much lively discussion and which concluded with software demonstrations from the participants.

The contribution by Del Gobbo and Mili employs relation-algebraic reasoning for composing the specification of a fault-tolerant flight control system, exploiting the flexibility of relational formalisations and the ease with which relational operations allow composition at a high level of abstraction to arrive at a logically structured specification that also addresses the non-functional requirement of fault-tolerance.

Also in the realm of specification, Khédri considers relational formalisations of scenarios. He arrives at a characterisation of bisimulation via fixpoints involving residuals, and uses this to outline a technique that can be used for automated comparison of different formalisations of

scenarios, yielding information not only about discrepancies, but also about different levels of formality and granularity.

Omodeo and Doberkat propose a relation-algebraic semantics for the static aspects of entity-relationship models with the aim of enabling formal reasoning about ER-models. They use a formalisation of set theory inside relation algebra to enable a formal semantics of a restricted kind of ER-models (but including inheritance) close to the intuitive understanding of ER-modellers.

A new kind of relational diagrams representing relation-algebraic formulae is introduced by Formisano, Omodeo and Simeoni, and implemented as an application of an existing graph rewriting tool. This provides a proof assistant for graphical reasoning in the calculus of binary relations, and includes mechanisms for the translation of predicate-logic specifications.

Bruni and Gadduci investigate fundamental properties of category-theoretic spans and co-pans and show how closely these relate to algebras of relations and of multirelations. An interesting application shows how co-pans, used as semantic domain for Petri nets, support reasoning about causality between different threads of a distributed computation.

Building on a relation-algebraic model of CCS processes, Winter presents an implemented mechanism for generating process definitions from safety conditions expressed in a modal logic, and includes a sketch how this could be extended to also cover temporal logic liveness conditions.

We are very grateful to Jules Desharnais (Université Laval, Québec, Canada) and Rudolf Berghammer (Christian-Albrechts-Universität zu Kiel, Germany) for joining us on the program committee for the workshop, and also wish to express our sincere gratitude to Michael Mislove for making this special issue of ENTCS possible.

Wolfram Kahl
David L. Parnas
Gunther Schmidt
Hamilton
Limerick
München
