ERCIM News No.23 - October 1995 - INRIA

Refining and Developing Concurrent Systems from Formal Specifications

by Dominique Méry and Denis Roegel

---

by Paul Gibson, Dominique Méry and Denis Roegel

Mathematical techniques have, in the last twenty years, been rapidly developing to improve the understanding of programs. Amongst these mathematical methods, proof techniques are considered very important, though difficult to apply in a practical manner. Industrial entities are aware that some (safety critical) software must be proved correct, requiring mathematical techniques and a formal background that is only partially provided by current training and education. Consequently, there is a drive towards the transfer of theoretical research into industrial usage. Proof techniques, in general, are very difficult to apply in industry. An alternative, yet equally powerful, solution is based on the notion of refinement relationship. A program P is said to be refined into another program Q, if any property satisfied by P is a property satisfied by Q. Properties such as safety, fairness or eventuality can be specified by a sufficiently expressive language, and a relationship between programs and such a specification language can be used to prove the correctness of programs with respect to such temporal requirements. A relationship between a program and a specification can be defined through an operational semantics for the program. When such a semantics exists, we can manipulate, in a formal manner, specification/program pairs. The objective of a refinement is to establish specification/program pairs and to produce transformations over programs, and/or specifications, that maintain the correctness of a program with respect to a specification.

What are the problems that remain to be solved? Clearly, we need to discover transformations that preserve specification/program pairs and to prove that classical transformations are sound according to our semantics. However, programming languages can deal with a wide range of conceptualisations such as tpyes, objects, databases, processes, functions, services, concurrency, distribution and safety critical requirements; moreover, a specification language may express many kinds of static properties (typing, invariance, deadlock freedom, partial correctness), dynamic properties (accessibility of critical section total correctness) or time-sensitive properties (liveness and fairness). Since any attempt to consider all those issues risks "missing the wood for the trees", we choose to limit the scope of the problem by considering particular case studies and developing solutions for specific classes of problem. In this way we hope to discover generally applicable models and methods of transformation. For example, one such domain which we are considering is the specification of services in a telecommunication system. The main problem is the ability to express composition of services whilst being aware of the problems due to feature interactions. Although this problem is specific to telecommunications, we believe that the mathematical means of solving the problem can be usefully applied in many other situations.

In practise, there are many different approaches being taken by industry in an attempt to rigorously satisfy constraints of integrity, safety, security, fairness and liveness. When such rigorous approaches are based on a language with firm mathematical foundations, then we refer to them as formal methods. The following formal methods are studied by

our research group:

- UNITY, designed by Chandy and Misra, deals with parallel program design.
- B, designed by J.R. Abrial, has been applied to a large range of industrial case studies.
- TLA+, designed by L. Lamport, gathers together many of the concepts seen in UNITY and B into a logic of temporal reasoning and set theory.

We aim to aid the transfer of formal methods to industry in two ways:

• Industrial Application: In the short term, direct co-operation between researchers and developers in an industrial setting must continue to transfer knowledge and understanding (in both directions). A balance must be achieved between showing the industrial designers what can be achieved with formal methods and adapting the mathematical methods to meet practical needs. Collaborations with industrial partners, for example, France Telecom and GEC Alsthom Transport are already underway.

• Education: In the long term, the future of formal methods in industry depends on both creating a demand for engineers trained in their use and providing a means of fulfilling this demand. It is a viscious circle: without the formally trained engineers it is difficult to create the demand (through suitable industrial applications and trials) yet without the demand there seems to be no reason to train the engineers. Our solution to the problem is to incorporate formality in our teaching and training as much as possible. As teachers, members of the group are involved in many training programmes and use them as an opportunity to spread the formal methods message. Consequently, many of our students are prepared for the application and integration of mathematical techniques when they take up positions in industry.

Our research group aims to improve the interaction between the industrial world and theoretical researchers, by promoting the use of formal methods in the design of programs, especially those with high levels of concurrency and distribution in safety-critical environments. We guide our research with the help of case studies, and construct tools that implement our theoretical results and aid case study development. Our goal is a set of tools which are formally based, powerful to apply and accessible to all software engineers at all levels of abstraction.

Please contact:
Dominique Méry - CRIN-CNRS & INRIA Lorraine
Tel: +33 8359 2014
E-mail: mery@loria.fr

---

return to the contents page