Full Synopsis

Teaching Formal Methods: Practice and Experience
Friday 15 December 2006

BCS London Office, 5 Southampton Street, London

This workshop was the second workshop to be organised by Oxford Brookes University's Department of Computing [1] and the BCS Formal Aspects of Computing Science Specialist Group [2] on the theme of teaching formal methods. The first workshop was held at Oxford Brookes University in December 2003 [1]. This first workshop was followed by a symposium organised by CoLogNET and Formal Methods Europe in November 2004 [4]. The Formal Methods Europe (FME) Education Group also organised a workshop on Formal Methods in the Teaching Laboratory as part of the FME 2006 symposium in Canada in August 2006 [5].

Teaching Formal Methods: Practice and Experience attracted 36 participants from around the world – countries represented included Argentina, Austria, Germany, Finland and Sweden. The programme included two invited contributions (Ralph-Johan Back, Åbo Akademi University, Finland and Rod Chapman, Praxis High Integrity Systems Ltd, UK) and thirteen refereed papers.

Ralph-Johan Back, whose attendance was sponsored by FME, opened the workshop with a stimulating talk on invariant-based programming. After outlining the approach he then spoke about experiences with teaching this approach and recent experiments in which small teams with no previous experience of the approach apply it to a variety of problems. There are plans to expand the teaching of this approach to high schools. The key advantage of the approach is that it allows programmers to check themselves that a program is working correctly.

The opening talk was followed by thirteen refereed papers organised into four sessions: tools in teaching, formal methods in the curriculum, teaching formal methods and tactics and techniques.

The papers by Broda et al, Schreiner, Feinerer and Back discussed a variety of tools that have been or are being developed to support teaching formal methods. Demonstrations of the tools were given during the lunch break. A book exhibition (publishers exhibiting: Palgrave Macmillan, Pearson and Springer) was also available during the lunch break and throughout the workshop.

Formal Methods in the Curriculum included a paper by Backhouse on experiences of teaching a module that introduces problem-solving skills based on the principles of correct-by-construction algorithm design. Backhouse spoke about the idea for this module at the TFM 2003 workshop; then it was an optional module, now it is a compulsory module for all first year students at the University of Nottingham – the paper reflected on the experience of teaching the module over 3 years and the transition from optional to compulsory status. Bayley's paper advocated a reverse engineering approach to teaching formal specification and Habrias described how a soroban (a Japanese abacus) and a Russian abacus and Chinese abacus are used to teach abstraction and refinement.

The *Teaching Formal Methods* session (Simpson, Cristiá, Loukanova) included reflection on the experience of teaching formal methods in the Software Engineering Programme at the University of Oxford – a specialist course aimed at part-time students many of whom are practising software engineers. The paper described a statistical analysis of results from the programme; questions studied included the extent to which performance in software engineering mathematics is a predicator of

overall performance in the course. Cristiá described a 1st semester year 4 course at U.N. de Rosari and U.N. de Córdoba, Argentina. The course is called Systems Analysis and aims to demonstrate that it is mathematics and logic that make software engineering and engineering discipline. The teaching style has evolved from slide-based presentations to sessions in which teachers solve concrete problems in class – class notes introduce notation as necessary through concrete examples. Experience has shown that it is abstraction and modelling that students find difficult, not logic or mathematics. Students who have taken the course and moved on to posts in industry find themselves dissatisfied with software development practices in industry and seek to apply the insights gained from the course in their industrial contexts. Loukanova's paper looks at teaching formal methods in a very different context – computational linguistics.

Tactics and Techniques included papers by Mosses, Boute and Bayley et al. Mosses described a course on semantics of programming languages he taught at the University of Aarhus before moving to Swansea. He demonstrated the advantages of Modular Structured Operational Semantics for this purpose – Modular SOS offers support for reusability (modules), there is no need for reformulation when extending modules, there is a clear distinction between abstract and concrete syntax and there is tool support available (currently in Prolog). Boute continued the semantics theme and argued that semantics is as important for computing as Kirchoff's laws are for electrical engineering. By introducing an elementary form of semantics (semantics of substitution) at a very early stage in the curriculum the foundations are put in place for introducing other concepts at a later stage. The paper by Bayley et al (presented by Lightfoot) reflected on the experience of teaching formal methods at Oxford Brookes University. The undergraduate module is based on Z, chosen for its constructive properties (e.g. sequence is a function, is a relation, is a set of pairs). A variety of case studies are used and some tool support is available. Use of tools is not required and in fact some benefits (namely in explanations that can be offered by tutors) have been found in not using tools.

The closing capstone talk was given by Rod Chapman of Praxis High Integrity Systems Ltd. He described experiences within his company in both the use and teaching of formal methods. Courses include SPARK at a variety of levels and Z. Rod stressed the importance of understanding the problem domain when developing specifications. Finally he offered some reflections on themes that had arisen during the day. He noted the need within industry for better tool support and encouraged academics to bring tools to market and to the wider attention of the industrial audience. He noted the need for tools to support impact analysis and change analysis (what are the consequences of changing this part of the specification?) and the relationship between the formal world and the real world in which integers are finite, floating point is faster than integer arithmetic and computers can fail in the worst way possible at the worst possible time.

The workshop concluded with a reception generously sponsored by the publishers Palgrave Macmillan, Pearson and Springer. Discussion continued well into the evening – a sure sign that delegates found the workshop stimulating and an excellent opportunity for discussing issues of common interest. Feedback received after the conference has been extremely positive.

The organisers would like to thank Escher Technologies for sponsoring the printing of the draft proceedings distributed at the event.

[1] hyperlink to http://www.bcs-facs.org/

[2] hyperlink to http://cms.brookes.ac.uk/computing/

[3] Proceedings of Teaching Formal Methods 2003, available at http://cms.brookes.ac.uk/tfm2003/

[4] C. Neville Dean, Raymond T. Boute (Eds.): Teaching Formal Methods, CoLogNET/FME Symposium, TFM 2004, Ghent, Belgium, November 18-19, 2004, Proceedings. Lecture Notes in Computer Science 3294 Springer 2004, ISBN 3-540-23611-2

[5] hyperlink to http://www.di.uminho.pt/FME-SoE/FMEd06/

Programme Committee