

GUEST EDITORIAL

Workshop on Foundations of Object-Oriented Languages

This issue contains papers selected from the sixth Workshop on the Foundations of Object-Oriented Languages (FOOL), held in San Antonio, California on January 23, 1999, in conjunction with the 26th ACM Symposium on Principles of Programming Languages (POPL '99). The workshop consisted of one invited lecture and seven contributed papers selected by the program committee. The four papers in this issue were selected from those presented at the workshop.

The FOOL workshops were initiated by Kim Bruce and Giuseppe Longo in 1993 with the assistance of grants from the NSF and ESPRIT to bring together researchers from around the world to discuss the theoretical foundations of object-oriented languages. Attendance at the first two workshops, held at Stanford University and CNAM in Paris, respectively, was by invitation only. Beginning with the FOOL 3 workshop, held in Brunswick, New Jersey, the workshops have been open to the public. FOOL 4, held in Paris in affiliation with POPL '97, attracted over 100 attendees. The most recent workshop, FOOL 8, was held in London in January 2001.

Specific topics addressed in the workshop have included semantics, foundational calculi, type theory, and program verification of object-oriented languages. More recently special efforts have been made to include contributions on foundational issues related to concurrent and distributed object-oriented languages, and database languages with object-oriented features. More information about the FOOL workshops, including reports on previous workshops, can be found at the url <http://www.cs.williams.edu/~kim/FOOL/>. This is the fourth special issue of a journal with selected papers from FOOL workshops. The first, which included papers from FOOL 3, appeared as volume 4, issue 1, of *Theory and Practice of Object Systems (TAPOS)* in 1998. The second, including papers from FOOL 4, appeared as volume 5, issue 1, of *TAPOS* in 1999. Starting with FOOL 5, the special issues started appearing in this journal. The third special issue, with papers from FOOL 5, appeared in a recent issue of *Information and Computation*. A forthcoming issue of *Information and Computation* will include papers from FOOL 7.

The first paper in this issue, "Aliasing Models for Mobile Objects," by Uwe Nestmann, Hans Hüttel, Josva Kleist, and Massimo Merro, discusses a formal model of object migration. Early work described migration as consisting of cloning an object and then using the original as an alias or proxy for the clone. The authors study this notion of object migration in a formal calculus, \mathcal{O} jeblik, which is intended to model the concurrent core of the language Obliq. The authors discuss four different aliasing models for \mathcal{O} jeblik. Interestingly the two models corresponding to existing formal semantics exhibit severe problems with respect to the transparency of migration. These problems are also reflected in implementations of Obliq. The authors show that one of the four models has much better behavior. This paper presents a good example of how formal analysis can help uncover problems with language designs and implementations.

The second paper, "Foundations for Virtual Types," by Atsushi Igarashi and Benjamin C. Pierce, presents a formal semantics of virtual types. Virtual types were first introduced in the languages Beta and have been proposed as an extension to Java. They provide a mechanism for increasing the expressiveness of object-oriented languages by providing greater flexibility in modifying types of methods and instance variables in subclasses. It may be seen as an alternative to parametric polymorphism for expanding the expressiveness of object-oriented languages. The authors present an encoding of virtual types in a functional setting in terms of dependent records with bounded type fields. The type theory necessary for this encoding is very complex, with extensions to imperative languages appearing even more problematic. These results suggest that parametric polymorphism may be a semantically simpler approach to adding this kind of expressiveness to object-oriented languages.

The third paper, "Foundations for Extensible Objects with Roles," by Giorgio Ghelli, is a contribution to the study of static type systems in object-oriented database systems. The main problem addressed is that over time an object may acquire added state and behavior without changing its identity. These

changes allow objects to take on new (perhaps incompatible) roles. To model this phenomenon Ghelli introduces a role calculus that is an extension of Abadi and Cardelli's object calculus. He presents a type soundness theorem which shows that the type system is safe with respect to an operational semantics.

The fourth paper, "Modular Statically Typed Multimethods," by Todd Millstein and Craig Chambers, addresses the well-known difficulty of performing modular type checking in a language with symmetric multi-methods. In general, type checking such a language requires whole-program analysis, as independent extensions of a program may individually be safe, but when combined can result in errors. The authors describe different restrictions that can ensure forms of modular type safety for languages with multimethods. Their work is performed in the context of a simple core language with multimethods, Dubious. The authors prove the type safety of each of these systems. This paper provides important insight into the trade-offs between expressiveness and modularity in systems supporting symmetric multimethods.

Overall these four papers reflect the important research that continues in the foundations of object-oriented programming languages.

Kim B. Bruce
Williams College
Guest Editor