Aliasing is naturally endemic in object-oriented programs. Every object in an object-oriented program can refer to many other objects, so the overall topology of a program is a general graph. The state of an object can rely on the states of all of the other objects it references. Aliasing occurs when one object is reachable via multiple paths, usually because it is referenced by several other objects. Aliasing results in "action at a distance"—operations on one object can affect the logical state of apparently unrelated objects that (perhaps indirectly) refer to it. The articles in this issue address the costs and benefits of aliasing in a number of ways.

The first three articles focus on programming techniques and language features that allow programmers to control aliasing and its effects. Boris Bokowski and Jan Vitek describe how Confined Types can be used to restrict potential aliasing within programs. John Boyland’s article on Alias Burying describes how variables holding unique references to objects can be implemented without special programming language support. Günther Kniesel and Dirk Theisen describe how Java can be extended to support types that provide read-only access to objects, thus eliminating some of the unwanted effects of aliasing without restricting the program’s topology.

While new languages and language features can help reduce and control aliasing, compilers today must deal with programs filled with potential aliases. The article by Anthony Hosking et al. describes how partial redundancy elimination can support compiler-level optimizations in the presence of aliasing. Finally, Olivier Zenda and Dominque Colnet evaluate the use of aliasing in the implementation of the SmallEiffel compiler.

This issue, then, represents the state of practice in dealing with aliasing in object-oriented systems. The issue grew out of a workshop on aliasing held in conjunction with ECOOP 1999 in Lisbon, Portugal; revised papers were submitted for inclusion in Software—Practice and Experience. We would like to thank all the reviewers for their care in considering the papers for this special issue: Paulo Sergio Almeida, Dave Detlefs, John Hogg, Rustan Leino, Eliot Moss, Aamod Sane, Raymie Stata, Alan Wills, and Phillip Yelland.

James Noble
School of Mathematical and Computing Sciences
Victoria University of Wellington
P.O. Box 600
Wellington
New Zealand

Doug Lea
Computer Science Department
State University of New York at Oswego
Oswego
NY 13126
U.S.A.