Object Structuring Techniques

John Potter
University of New South Wales, Australia

The aim of this tutorial is to present practical techniques for structuring objects, with a particular focus on implementation strategies for making object structure explicit. Programming language support for encapsulation is primarily provided at the class or module level. Design methods merrily talk of object composition and aggregation, but there is little effective support provided in programming languages for direct representation of these abstractions in code. Typically composite structures are represented through method calls for constructing objects. The tutorial will present the notion of object ownership, its underlying theory based on the graph theoretical notion of domination, and programming idioms for representing and maintaining ownership.

Outline of Tutorial:
- Review of composition and aggregation at the design level. Standard implementation strategies and typical problems.
- Introduction to the concept of object ownership. Theory of object ownership and containment invariants based on the graph theoretical notion of domination.
- Practical implementation techniques for structured objects. Various idioms will be presented in different languages, including Eiffel, Java and C++.
- Examples of how to exploit object ownership to provide customized copy and equality tests that trace object references to a natural depth. Object ownership can also be exploited to provide improvements to memory management.
- A system for dynamic visualization of owned objects will be demonstrated.
- An overview of ownership types, and how incorporating ownership types into programming languages and compilers would ease the management of ownership.

Who should attend:
Software architects, designers and programmers with a good degree of experience in object-oriented programming techniques.

John Potter is currently an associate professor in Computer Science and Engineering at the University of New South Wales, Sydney. Prior to that he headed up the Object Technology Group at the Microsoft Research Institute in Sydney. His teaching and research interests are focused on software design and programming languages. In addition to object technology, he is interested in component-based software, declarative programming languages and formal methods.

John has been working with objects for more than a decade. He brought Eiffel into the undergraduate curriculum at an early stage. He seeks to invent notations, tools and techniques that make it easier for developers to generate correct and efficient code.

He was the founding chair of the TOOLS Pacific conference series in 1990, filling that position for 3 consecutive years.