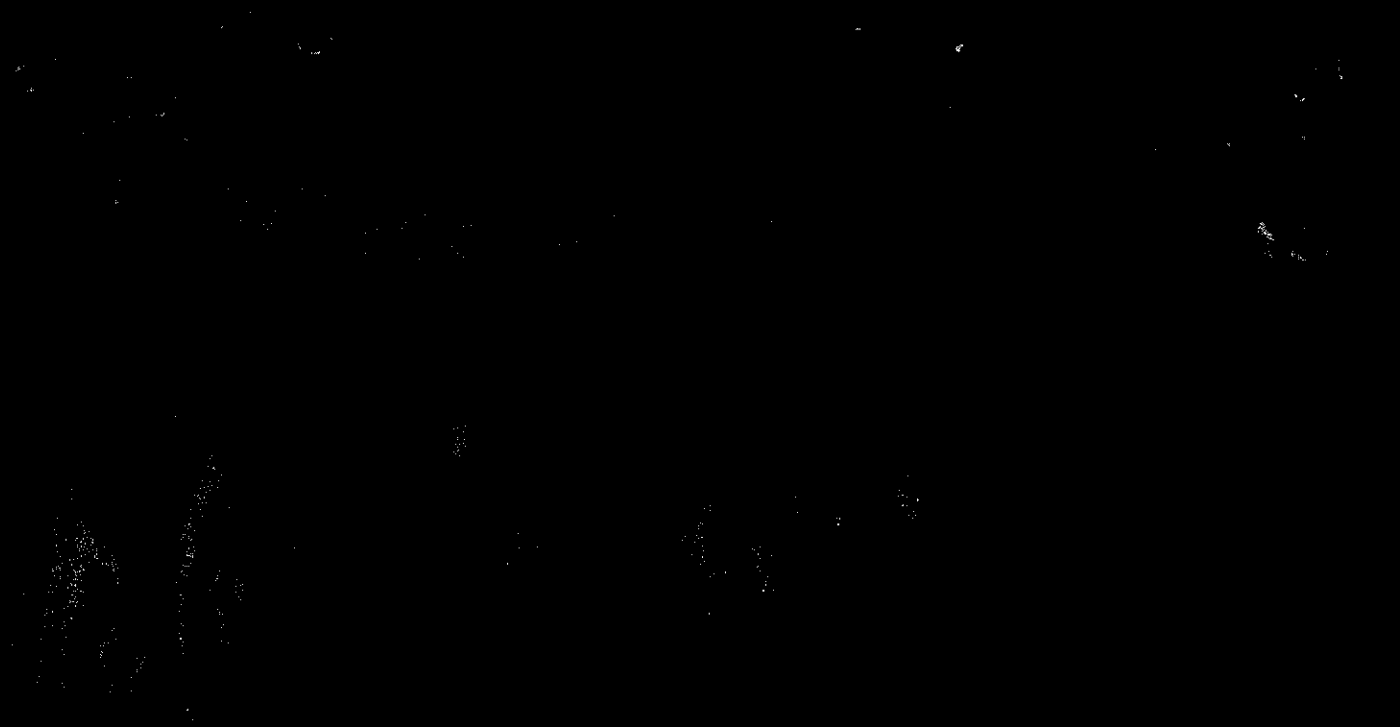


Patron Club and Hotel, San Francisco

JIRI SOURUP



TAMING C++

Pattern Classes and Persistence for Large Projects

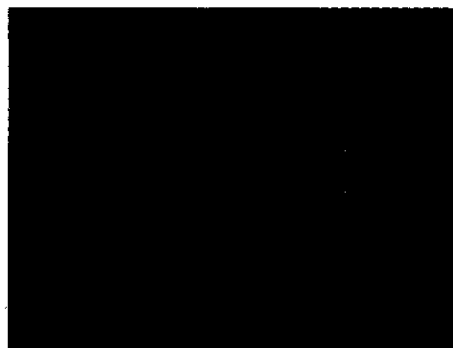
Jiri Soukup

The relationships and cooperation of C++ classes are central issues in large project development, in testing, and in maintenance. *Taming C++* takes a fresh look at the complex organization typical for large C++ systems and shows how this problem can be minimized by avoiding cyclic dependencies between classes, and by implementing the original object-oriented design so that it remains visible in the final code. This book shows you, through extensive examples, how to design and write C++ code using a new type of class—the pattern class—which helps you produce layered class organization with limited cyclic dependencies. The book then focuses on another problem of large project development—persistent data—integrating it with its general approach and providing copious implementation details unavailable elsewhere.

Highlights

- Emphasizes code readability achieved through better class organization
- Teaches how to improve software architecture
- Shows how pattern classes encapsulate the high-level object-oriented design
- Promotes layered structuring of class relationships
- Compares and contrasts design methodologies, class libraries, approaches to testing, and treatments of persistence
- Explains how persistent data can be used as fast databases (frameworks) residing in virtual memory

Taming C++ is written for programmers, designers, and software managers. It will also be useful to applications developers, researchers, and advanced students. While the techniques presented here can improve any application, they are specifically important for large projects.



About the Author

Dr. Jiri Soukup is President of Code Farms, Inc., a company promoting better ways to design software. Its product, the C++ Data Object library, has built-in pattern classes and automatic persistency. Dr. Soukup was one of the key contributors to several state-of-the-art CAD projects. He was director and a founding member of Cadence Design Systems, manager of AT&T's VLSI layout system, and one of the key developers of a printed board layout system currently sold by IBM.

Programs from this book can be accessed via anonymous ftp at: `ftp aw.com` under `aw.computer.science` in a file entitled `soukup`; the code is also available on disk from Addison-Wesley Publishing Company; send requests to `soukup@aw.com`. Comments to the author can also be sent to this same address.



9 780201 528268

ISBN 0-201-52826-6



Addison-Wesley Publishing Company

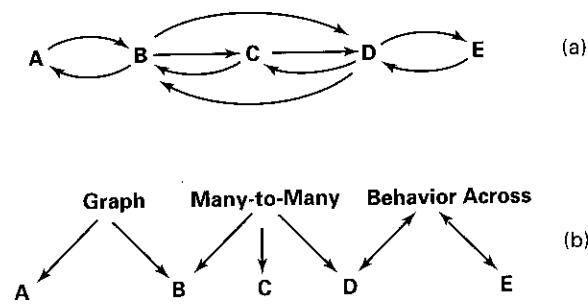


Figure 2.18 Two implementations of three overlapping patterns: (a) without pattern classes, (b) with pattern classes. Arrows represent friend relations. Compare the chaotic situation in case (a) where every class depends on every other class, with the orderly case (b) where application classes remain independent.

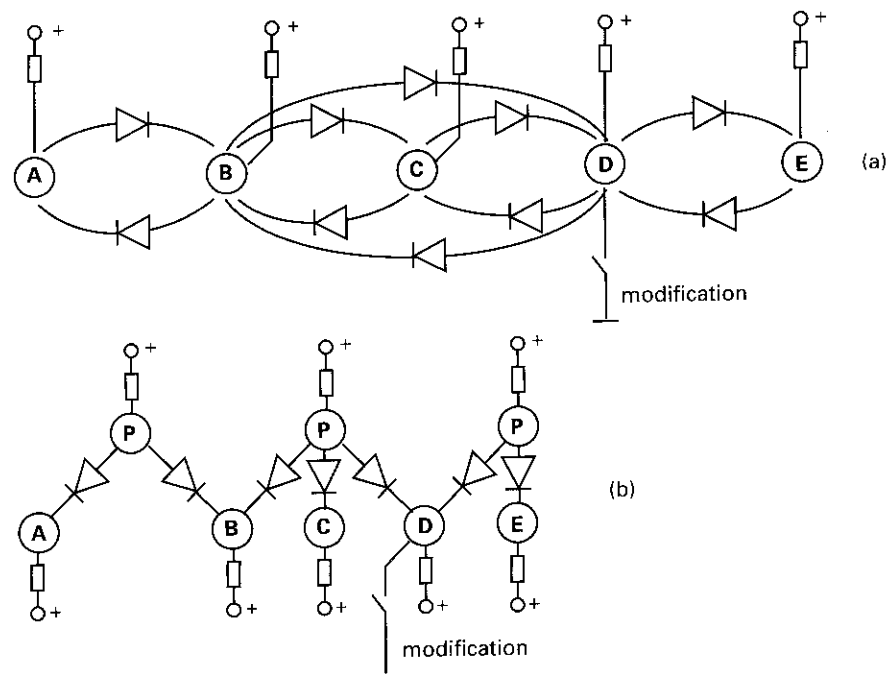


Figure 2.19 Electrical analogy of the situation in Figure 2.18. Classes are represented as nodes kept at high voltage, diodes represent friend relations. When a voltage drops, the class must be modified. The analogy illustrates how, without pattern classes, a change propagates to all classes, case (a). When pattern classes are used, only the relevant pattern classes are affected, case (b).