Object Engineering

Designing Large-Scale Object-Oriented Systems

Gary C. Sullo

Object Engineering

Designing Large-Scale, Object-Oriented Systems

Gary C. Sullo

A Wiley–QED Publication

JOHN WILEY & SONS, INC.

New York • Chichester • Brisbande • Toronto • Singapore

Library of Congress Cataloging-in-Publication Data:

Sullo, Gary, 1953-
Object engineering: designing large-scale, object-oriented systems/Gary Sullo.
p. cm.
Includes index.
ISBN 0-471-62369-5
1. Object-oriented programming (Computer science)
2. System design. I. Title.
QA76.64.S83 1994
005.1'2—dc20 94-1968
CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Object Engineering

Preface

As a software engineer on real-world projects, like many of you, it seems that I am continually faced with learning new ways of looking at my craft and new ways of applying my skills to build systems and to advise others in doing the same. Structured design was good, and it helped to build better systems (sometimes). Then information engineering was introduced, and it was sometimes better for building other kinds of systems. A lot to learn, but it seemed like this covered it all, and in many ways, it did. What, then, is object-oriented design all about? An object is just another way of looking at a component of a system, like the process modules in structured design or the data entities in information engineering. This is good news. It means that your prior knowledge is useful after all.

Object-oriented programs are said to be easy to use. Very easy. Why then, must their design be so complicated? The answer, of course, is that it doesn't have to be. Surely, something as popular as object-oriented design can be explained without requiring that we totally retrain our professional work force. And that is the purpose of this book—to explain object-oriented design in the context of the things that you already know (which is how we all learn it, anyway) and, as a result, to help you to upgrade your skills with the least amount of pain this time. This shouldn't be the only book that you read on object-oriented design, but it should be one of them.

This book is for those professionals and students of computer science who are familiar with software engineering methodologies and who have been exposed to objects, object languages, and object tools, but who need

to have the smoke cleared away. This will include software engineers, programmers, analysts, and systems integrators. The software engineers and systems integrators who will benefit from reading this book are those who are charged with designing, managing, or integrating large scale industrial strength systems using object-oriented technology. The programmers and analysts who will benefit are those charged with carrying out those projects.

Object engineering doesn't rely on any one particular technology, but rather gives you the understanding of how that technology fits into your project. If you are interested in keeping your skills up to date and need to have all of the new terminology put into a more meaningful perspective, then this book is for you. It will :

- Define the terms used in object-oriented design in a clear and concise manner, so as to bring them together in a meaningful way.

- Relate object-oriented design to concepts borrowed from conventional design, which are more familiar to you anyway.

- Lay out an organized model and a corresponding methodology for object-oriented design which is both comprehensive and yet flexible.

- Provide cross references to the many other object-oriented notations, diagrams, and techniques commonly used in the industry.

Object engineering is a methodology for designing large-scale, object-oriented systems. This book explains *the methodology* in a way that eases your transition into the object-oriented world. The book builds on what you already know about system development. Part I reviews the principles of conventional software design as the context for designing objects. Part II then defines the specific components of an object-oriented design within that context. Finally, Part III provides a layered model for developing that design. The book tells a building story from front to back, but you can also skip around from topic to topic without any loss of continuity. It is, therefore, both a training guide and a reference book.

By approaching the topic in this way, the book helps you to understand that object-oriented design is an evolution, rather than a revolution, in software engineering. This is a unique approach to teaching object-oriented design techniques, and yet, results in a model which is still consistent with the many other diagram notations and design methods that you may encounter elsewhere. Whether you are using Booch, Coad/Yourdon, or some other specific set of diagrams and techniques, and whether you favor CORBA, IEEE, Microsoft, or some other set of object standards, the principles of object engineering should help you to move more quickly and to be more effective in the new world of object-oriented design.

Gary C. Sullo

Object Engineering

Contents