



Software Systems Construction with Examples in Ada

Bo Sanden

Software Systems Construction with Examples in Ada is written by Bo Sanden, and published by Prentice-Hall, Inc., 1994, ISBN 0-13-030834-X, 433 pages, \$ unknown by reviewer

This book certainly lives up to its title. The author delivers a text "intended for readers with practical experience in software construction. It targets graduate or upper-level undergraduate students of software engineering as well as practitioners who want to familiarize themselves with modern trends in software construction applied in a mainstream language environment." As the author notes, the book could be complimented by an Ada primer and a software engineering text. This reviewer found the text well structured, filled with appropriate references, expanded with a number of design examples, and supported by a large amount of Ada source code fragments (specs and bodies). Additionally, each chapter concludes with a number of problem statements. I am sure with the author's attention to detail, an updated edition covering Ada95's new object oriented (OO) and real-time features will be forthcoming.

The text includes nine chapters that cover software construction in Chapters 2-7, present entity-life modeling in Chapter 8, and conclude with Chapter 9 by walking through a case study of a flexible manufacturing system from problem statement to Ada implementation.

Discussions on software construction build through the text and cover different aspects such as control structuring, modularization, information hiding, abstract data types, finite automata, concurrent tasks, and resource sharing. Chapter 8 presents entity-life modeling as an approach for development of concurrent programs. The discussion builds nicely on the earlier topics and covers the properties of concurrent programs. Then moves on to the analysis and design methods that support the modeling approach. Two well-known problems (the buoy and elevator examples) are provided with their ensuing entity-life solutions. Chapter 9 provides a detailed case study by applying the modeling approach to a non-trivial flexible manufacturing system. This culminating chapter also serves to tie all of the earlier discussion together.

I would strongly recommend this book for a curriculum that covers software construction and the Ada language. The introduction of entity-life modeling broadens the target community to the OO realm; however, the experienced OO practitioner may need only to focus on the final two chapters.

Reviewed by Frank Hollenbach —
frankhol@nadc.nadc.navy.mil.

An ISO 9000 Approach To Building Quality Software

Östen Oskarsson and Robert L. Glass

An ISO 9000 Approach To Building Quality Software is written by Östen Oskarsson and Robert L. Glass and is published by Prentice Hall PTR 1996, hardbound, ISBN 0-13-228925-3, 274 pages, \$49.95.

In this self-styled "schizophrenic" analysis of the ISO 9001 and 9000-3 standards, Oskarsson and Glass take contrasting (but not incompatible) positions on the standard's value for engineering quality software. In Part 1, Oskarsson is primarily concerned with exegesis and application of ISO 9001 - how to interpret the production-oriented standard into the design-oriented world of software engineering and how to apply the standard in the creation of a certifiable quality system. In Part 2, Glass compares the stipulations of the ISO 9000-3 standard with the tools and techniques available for software quality assurance and repeatedly concludes that the ISO document, *qua* quality "procedure" (in the terminology of auditing), is deficient.

In his monograph, Oskarsson takes the conventional auditor's view that ISO 9001 stipulates the minimal *management* requirements for a quality system. He explains the context in which the ISO standards were developed, introduces some of the terminology of audits and certification, and then explains each of the 20 ISO 9001 elements in turn. To alleviate the tedium of the exegesis, Oskarsson uses anecdotes from his auditing experiences to illustrate the standard's requirements and to identify those policies and procedures that are particularly prone to non-conformance citations. He ends with a brief summary of the necessary components of a quality system and the steps along the way to certification.

Much of the current ISO literature is content to parrot the standard; Oskarsson interprets it using his TickIT expertise and auditing experience, examining fine distinctions and problematic cross-references that other works ignore. (E.g., what are the implications of the traceability and inspection requirements for testing policies?) Readers interested in detailed analysis of what ISO 9001 says and doesn't say would be better served by struggling through Allan Sayle's *Meeting ISO 9000 in a TQM World*, and those concerned with how to get a certificate will find Charles Schmauch's *ISO 9000 For Software Developers* and Lawrence Wilson's *Eight-Step Process to Successful ISO 9000 Implementation* more useful. But as an *overview* of the *standard*, Oskarsson's discussion is unparalleled — just the sort of thing one would want to hand to one's engineering manager when s/he comes by and says "We've been told we have to go for ISO. What's that mean?"

Glass's Part 2 will disappoint readers who view ISO 9001 and ISO 9000-3 as an auditing standard, a verbose checklist for assessing project management capability. For one hundred and fifty pages he seems to belabour the obvious: the standards specify a quality management system, they don't guarantee quality software; they address the principles used to manage