

ant. But they note that these Contracts exist at different levels, depending on the intention of the model. Thus, "A contract is a *specification* for all its lower levels and a part of the *implementation* of all its higher levels." (p 31)

The authors confusingly use *object* and *entity* sometimes as synonyms, while at other times they offer key distinctions between older approaches (based on entity-relationship modeling) and their *association-of-objects*-oriented approach. Indeed, they suggest that we use *associations* to discover objects. They offer a chapter which contains descriptions of a library of generic associations – which we might characterize as 'patterns' in current jargon – useful across an array of individual instances of information models. These generic associations include, for example, Dependency, Reference, and several variations on Composition.

One of the feature of the book I found most appealing occurs in the chapter containing guidelines. Here they make excellent use of an extended example (14 pages). They frame the example as a dialog between a modeler and a Subject Matter Expert (SME) incrementally constructing an information model through a dialog. A key feature of this example involves having the modeler think aloud and explain the motives for the questions posed! Getting inside the head of the modeler by having that modeler make *explicit* some of those *implicit* thought processes serves as key part of discovering and promoting those 'good thinking' practices mentioned on the first page. They recognize that we will not find many situations as tidy as their example. "Understanding the business often takes a long time: not all business are like the little enterprise shown in our example! It is not uncommon to have modeling [efforts] that take several months. This should be explicitly recognized, because often there seems to be no 'visible' progress in modeling. . . . This happens especially when the modeler asks hard questions, that is, questions for which the SME does not have an immediate definite answer. These questions uncover some business rules that may never have been explicitly stated." (p 159)

Still, they contend that we must take this rigorous sort of approach. "To make information modeling a discipline, we must expose some patterns of reasoning that apply to information modeling. According to Dijkstra, such exposure is essential for moving from craft to a more disciplined approach." (p 166) And only through achieving a discipline can we raise awareness of the priority of analysis in the scheme of developing information systems.

Reviewed by Michael Ayers, 3M Center, IT Educ Svcs, 224-2NE-02, St. Paul, MN 55133, 612.733.5690 (voice) 612.737.7718 (fax), mbayers@mmm.com.

## PCTE: The Standard for Open Repositories

Lois Wakeman and Jonathan Jowett

*PCTE: The Standard for Open Repositories* is written by Lois Wakeman and Jonathan Jowett, and is published by Prentice Hall International (UK) Limited, 1993, Paperback, ISBN Number 0-13-065566-x, 262 pp., Cost \$51.00.

PCTE: The Standard for Open Repositories provides an overview and examples of the Portable Common Tool Environment (PCTE), which is a public tool interface (PTI) for an open repository. The book provides a summary of the two part structure of the specification set: the abstract specification Standard ECMA-149; and the binding specifications, such as ECMA-158 for C and ECMA-162 for Ada.

The book is organized for use as a reference, for either an informational overview or an initial detail level about the actual implementation of PCTE. There are seven chapters discussing different aspects and information about PCTE and its implementation at a high level, two appendices which discuss the history and data definition conventions, a glossary, a bibliography and an index.

If you or your organization is interested in developing or utilizing aspects of PCTE to support reuse of existing or new products and services for your application domain, this is a useful reference for all interested personnel. To utilize this book, being familiar with domain architectures, open system architectures, and object oriented methods and techniques is not necessary, but does help to understand some of the notations.

The use of examples, figures, note boxes and tables was appropriate with the narrative text of the book itself. PCTE may not be fully implemented for all application domains, but many good starting point examples are given to structure your own or your organization's existing capital resources towards a PCTE.

The only shortfall of the book is that it clearly states that PCTE is NOT for proprietary or de facto standards, only for recognized publicly available or de jure standards, such as IEEE or ISO.

I found the book to be useful for clearly identifying from the beginning that it was an overview of the PCTE standards set and not all information available. With the examples, figures, note boxes and tables, an individual or organization can determine how PCTE may or may not be for their application domain.

Reviewed by David John Leciston, CECOM RDEC SED, Building 1210, Room 223 Rittko Avenue, Fort Monmouth, New Jersey 07703-5207, Voice: (908) 532-0700, Facsimile: (908) 532-8287, leciston@doim6.monmouth.army.mil.