Dines Bjørner

Software Engineering

Texts in Theoretical Computer Science, an EATCS series.

Due Spring 2005

**Vol. 1 – Abstraction and Modelling**
ISBN 3-540-21149-7
Approx. 730 pp., Euro 74.95, £57.50, $89.95

**Vol. 2 – Specification of Systems and Languages**
ISBN 3-540-21150-0
Approx. 780 pp., Euro 74.95, £57.50, $89.95

**Vol. 3 – Domains, Requirements, and Software Design**
ISBN 3-540-21151-9
Approx. 770 pp., Euro 74.95, £57.50, $89.95

The art, craft, discipline, logic, practice, and science of developing large-scale software products needs a believable, professional base. The textbooks in this three-volume series combine informal, engineeringly sound approaches with the rigor of formal, mathematics-based approaches.

**Volume 1** – The basic principles and techniques of abstraction and modelling. This volume is suitable for university undergraduate students and college lecturers.

**Volume 2** – The basic principles and techniques of specifying systems and languages. This volume is suitable for late undergraduate to early graduate university students, college lecturers, and researchers of programming methodologies. [Vol. 1 of this series is a prerequisite text.]

**Volume 3** – The basic principles and techniques of overall software development, from domains via requirements to software designs, advocating a novel approach to software engineering based on the adage “Before requirements can be formulated one must understand the application domain.” There are two reading paths: focusing only on the informal parts, this volume targets undergraduate students in courses on software engineering and college lecturers in that field, while the full version of the volume targets advanced students, lecturers, and researchers.

**Course Supports** – The volumes are supported by a comprehensive guide on their use in various dedicated courses, there are thousands of teaching slides, and extensive solutions have been worked out for hundreds of assignments posed in the volumes. Information can be obtained at: [http://www.comp.nus.edu.sg/~bjorner/Software_Engineering](http://www.comp.nus.edu.sg/~bjorner/Software_Engineering).
**Volume 1** (contd.) – First this volume provides a sound, but simple basis of insight into discrete mathematics: numbers, sets, Cartesians, types, functions, the Lambda Calculus, algebras, and mathematical logic. Then it teaches and trains its readers in basic property- and model-oriented specification principles and techniques. The model-oriented concepts that are common to such specification languages as B, VDM-SL, and Z are propagated here through the use of the RAISE specification language (RSL). Finally this volume covers the basic principles of functional, imperative, and parallel speciﬁcation programming, the latter based on the use of CSP, Hoare's language of communicating sequential processes.

**Volume 2** (contd.) – First this volume teaches and trains its readers in basic and advanced principles and techniques: hierarchical vs. compositional, denotational vs. computational, and conﬁgurational (abstracting and modelling contexts and states). Then it covers the basic principles and techniques of modelling the semiotics: pragmatics, semantics, and syntax of systems and languages. It then covers principles and techniques of modelling spatial and simple temporal phenomena, and such specialized topics as modularity (incl. UML class diagrams), Petri nets, live sequence charts, statecharts, and temporal logics, including the Duration Calculus. Finally this volume presents principles and techniques for developing the basis for sound, efficient interpreter and compiler development of functional, imperative, modular, and parallel programming languages.

**Volume 3** (contd.) – This volume is structured thus: from the principles and techniques for the development of domain descriptions, via principles and techniques for the derivation of requirements prescriptions from domain models, to principles and techniques for the reﬁnement of requirements into software designs, i.e., architectures and component design. Emphasis in the coverage of domain and requirements engineering is on what goes into a proper domain description (resp., requirements prescription), how one acquires and analyzes the domain knowledge (resp., requirements expectations), and how one validates and veriﬁes domain (resp., requirements) models.