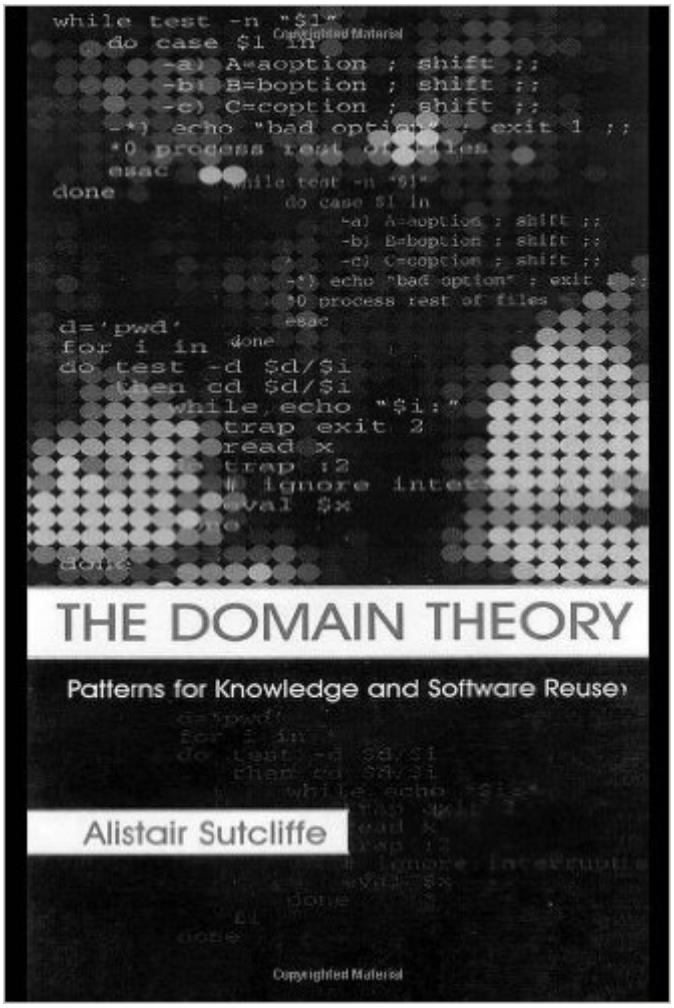
The Domain Theory: Patterns For Knowledge And Software Reuse



Synopsis

Is this book about patterns? Yes and no. It is about software reuse and representation of knowledge that can be reapplied in similar situations; however, it does not follow the classic Alexandine conventions of the patterns community--i.e. Problem- solution- forces- context- example, etc. Chapter 6 on claims comes close to classic patterns, and the whole book can be viewed as a patterns language of abstract models for software engineering and HCI. So what sort of patterns does it contain? Specifications, conceptual models, design advice, but sorry not code. Plenty of other C++ code pattern books (see PLOP series). Nearest relative in published patterns books are Fowler's (1995) Analysis Patterns: Reusable object models and Coad, North and Mayfield. What do you mean by a Domain Theory? Not domains in the abstract mathematical sense, but domains in the knowledge--natural language sense, close to the everyday meaning when we talk about the application domain of a computer system, such as car rental, satellite tracking, whatever. The book is an attempt to answer the question ' what are the abstractions behind car rental, satellite tracking' so good design solutions for those problems can be reused. I work in industry, so what's in it for me? A new way of looking at software reuse, ideas for organizing a software and knowledge reuse program, new processes for reusing knowledge in requirements analysis, conceptual modeling and software specification. I am an academic, should I be interested? Yes if your research involves software engineering, reuse, requirements engineering, human computer interaction, knowledge engineering, ontologies and knowledge management. For teaching it may be useful for Master courses on reuse, requirements and knowledge engineering. More generally if you are interested in exploring what the concept of abstraction is when you extend it beyond programming languages, formal specification, abstract data types, etc towards requirements and domain knowledge. ADDITIONAL COPY: Based on more than 10 years of research by the author, this book is about putting software reuse on a firmer footing. Utilizing a multidisciplinary perspective--psychology and management science, as well as software--it describes the Domain Theory as a solution. The domain theory provides an abstract theory that defines a generic, reusable model of domain knowledge. Providing a comprehensive library of reusable models, practice methods for reuse, and theoretical insight, this book: *introduces the subject area of reuse and software engineering and explains a framework for comparing different reuse approaches; *develops a metric-oriented framework to assess the reuse claims of three competing approaches: patterns, ERPs, and the Domain Theory OSMs (object system models); *explains the psychological background for reuse and describes generic tasks and meta-domains; *introduces claims that provide a representation of design knowledge attached to Domain Theory models, as well as being a schema for representing

reusable knowledge in nearly any form; *reports research that resulted from the convergence of the two theories; *describes the methods, techniques, and guidelines of design for reuse--the process of abstraction; and *elaborates the framework to investigate the future of reuse by different paradigms, generation of applications from requirements languages, and component-based software engineering via reuse libraries.

Book Information

Hardcover: 424 pages

Publisher: CRC Press; 1 edition (March 1, 2002)

Language: English

ISBN-10: 0805839518

ISBN-13: 978-0805839517

Product Dimensions: 6.7 x 1.1 x 8.8 inches

Shipping Weight: 1.5 pounds (View shipping rates and policies)

Average Customer Review: 5.0 out of 5 starsÂ Â See all reviewsÂ (1 customer review)

Best Sellers Rank: #4,508,772 in Books (See Top 100 in Books) #34 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Reuse #1545 inÂ Books > Computers & Technology > Databases & Big Data > Data Mining #3065 inÂ Books > Textbooks > Computer Science > Database Storage & Design

Customer Reviews

In this book, Alistair Sutcliffe steps back and examines reuse conceptually and soberly from a broad historic and analytical framework. He puts the reuse discipline on firm ground. He takes our hand, steps back and asks us to take a tour of the broader landscape. How has reuse evolved in other disciplines? He explores this question not by hollow or superficial analogy, but with a constant subtext that rigorously grounds the necessary arguments and concepts. What, exactly, are we trying to reuse, and where does the most payoff lie in reuse if it indeed does exist? Perhaps it isn’t in the reuse of artifacts at all, but in the reuse of knowledge. Even the reuse of data and interpreted data--information--has value. We can dare talk about wisdom reuse. Knowledge reuse itself is a broad topic ranging from rules and heuristics to procedures and propositions and, yes, even to designs. And reuse is an interdisciplinary phenomenon that builds on the business considerations one finds in enterprise modeling, the economic analyses of investment, the social analyses of process, the psychological models of conceptualization, as well as the technical models of design and technology. It is a broad and refreshing tour that reveals a landscape that surrounds all of us,

but which few are able to see.

The Domain Theory: Patterns for Knowledge and Software Reuse Software Reuse for Dynamic Systems in the Cloud and Beyond: 14th International Conference on Software Reuse, ICSR 2015, Miami, FL, USA, January 4-6, ... (Lecture Notes in Computer Science) Safe and Secure Software Reuse: 13th International Conference on Software Reuse, ICSR 2013,Pisa, Italy, June 18-20, 2013, Proceedings (Lecture Notes in Computer Science) Software Reuse Techniques: Adding Reuse to the System Development Process Reuse of Off-the-Shelf Components: 9th International Conference on Software Reuse, ICSR 2006, Torino, Italy, June 12-15, 2006, Proceedings (Lecture Notes in Computer Science) Domain Names For Profit: How to Play The Domain Name Game & Make Money Integrated Theory & Knowledge Development in Nursing, 8e (Chinn, Integrated Theory and Knowledge Development in Nursing) Software Reuse: Guidelines and Methods (Software Science and Engineering) Reengineering Software: How to Reuse Programming to Build New State-of-the-art Software Practical Software Reuse (Wiley Series in Software Engineering Practice) Object-oriented software development: Engineering software for reuse Software Reuse: Advances in Software Reusability: 6th International Conference, ICSR-6 Vienna, Austria, June 27-29, 2000 Proceedings (Lecture Notes in Computer Science) Software Reuse: A Holistic Approach (Wiley Series in Software-Based Systems) IntAR, Interventions Adaptive Reuse, Volume 03; Adaptive Reuse in Emerging Economies Domain-Driven Design: Tackling Complexity in the Heart of Software Applying Domain-Driven Design and Patterns: With Examples in C# and .NET Reusing Open Source Code: Value Creation and Value Appropriation Perspectives on Knowledge Reuse (Innovation und Entrepreneurship) Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection: Obfuscation, Watermarking, and Tamperproofing for Software Protection Software Engineering Classics: Software Project Survival Guide/ Debugging the Development Process/ Dynamics of Software Development (Programming/General) Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures