Architecting With RM-ODP is written by Janis Putman in English language. Release on 2001, this book has 834 page count that attach valuable information with easy reading experience. The book was publish by Prentice Hall Professional, it is one of best computers book genre that gave you everything love about reading. You can find Architecting With RM-ODP book with ISBN 0130191167.

Preface To understand anything, you should not try to understand everything. -- Aristotle
The whole is greater than the sum of the parts; the part is greater than a fraction of the whole. -- Aristotle Architecting is a challenging process of abstraction, composition, modularity, and simplification to create an architecture specification. An architecture specification captures the essence and definition of the system: understanding, parts, and the relationships among the parts. An architecture specification defines how a system solves a business problem within the scope of the business. -- Putman
Leave the beaten track occasionally and dive into the woods. You will be certain to find something that you have never seen before. -- Alexander Graham Bell There are large gaps in the theory and practice of software architecture and engineering. Much is published about
the representation of a software architecture, such as the Unified Modeling Language (UML), but little is available about the specification for a software architecture. Software engineering methods of domain engineering, process modeling languages, and well-formed patterns of reasoning aid in the specification of an architecture. The Reference Model of Open Distributed Processing (RM-ODP) defines the standard reference model for distributed software systems architectures, based on object-oriented techniques, accepted at the international level. RM-ODP is a standard adopted by the International Standards Organization (ISO) and the International Telecommunications Union (ITU). RM-ODP is embedded and used actively in mission-critical systems industries such as in telecommunications, in health care, on Wall Street (financial services industry), in various Government systems (Logistics), in European Government Agencies such as UK Aviation control systems, as a foundation for the Object Management Group (OMG) Object Management Architecture (OMA), for defining enterprise architectures, and for defining software architectures. The software systems architecture work that is emerging, and is focused either at the component level or at the systems level, provides a key resource for architects. This is enhanced by the architecting techniques of RM-ODP. This book assembles these great ideas, explains what they mean, and shows how to use them for practical benefit, along with real-world case study examples. By using the RM-ODP specification constructs, associated languages, architecture patterns of reasoning, semantic behavior specification, and conformance testing abilities, readers will be able to architect their specific systems based on the RM-ODP specification foundations, and specify architectures that work.

One of the purposes of this book is to provide the approach to using the RM-ODP foundations in architecting and specifying a distributed processing system that addresses such key properties as interoperability, dependability, portability, integration, composability, scalability, transparency, behavior specification, quality of service, policy management, federation, and conformance validation. Another purpose of this book is to explain the underlying foundations for creating an architectural specification. These foundations come not only from RM-ODP, but also from the current work in software systems architecture. Another purpose is to guide the reader to understand the importance and benefits of creating an architecture specification for an enterprise. Yet another purpose is to provide the reader with the principles to construct software systems architecture (at both introductory and in-depth levels). By applying the proven techniques of RM-ODP for what makes a good architecture, readers will be able to build their own tailored architectures, and clearly represent them in UML or some other tool, with an understanding of the underlying principles. Practitioners of RM-ODP have found that the standard is extremely beneficial in guiding architecture definition and providing standard terminology/principles for distributed object applications and infrastructures from an enterprise perspective. Outstanding Features This book is intended to provide valuable insight into successful architecture specification by describing an unprecedented foundation to accomplish this task, describing the use of the foundation, explaining the relationships of the concepts of architecting, explaining the relationships of the concepts of distributed processing, and identifying the right methods and possible tools for architecting. All material for the book has been derived from actual experiences. A medical case study is used throughout the book in ever increasing detailed specification. This medical case study is based on actual experience of the author. In addition, many metamodels are provided to represent the concepts of RM-ODP. All of these metamodels are contributions from the author. This is information that readers can use and apply in their architecting today. RM-ODP provides a reference framework, grammars, methods of abstraction and composition, and separation of concerns to achieve an architecture specification of the system. RM-ODP provides a framework for this separation, using viewpoints, as well as separating out certain decisions (e.g., product decisions) until later. Further, the reference model provides a set of terms for communicating among parties. There are no standards about RM-ODP or architecture specification, and certainly not a book dedicated as a tutorial of these subjects. Now there is. In summary, this book offers the following: How to manage the architecting process in the lifecycle of a system How to solve many architecture reuse and cost-effectiveness problems How to create a business specification How to understand and use the concepts of distributed processing in an architecture How to architect effectively How to specify an architecture How to architecting endeavors. It is the formalized framework for this topic, and at the International Standard (IS) level of acceptance. It forms a solution set and foundation for reuse of design patterns to provide cost-effective software architecture. It is the process for this topic, but has never before been described in a book. Many program managers (who typically set the stage as to the methodology of choice for a project), software engineers, and researchers in academia and in DARPA are unaware of the power and solutions provided by the standard, or the process of identifying and instantiating reuse of all the expensive assets of architecture. Many do not realize that there is a language for specifying software-intensive distributed processing, and that language is precisely and rigorously defined in RM-ODP for reuse. Those debating definitions for architecture, system, interface, and others can reuse the internationally agreed upon definitions. Finally, with the inclusion of RM-ODP and its relationship to
other architecture frameworks, it is expected that many software engineers will benefit from reading this work, since it will be the first time these subjects are discussed in print. How to Use This Book This book is divided into four parts, aimed at increasing levels of detail. Part One provides an overview of the field of software architecture, an RM-ODP primer for managers, and an RM-ODP primer for architects. Part Two provides an in-depth study of RM-ODP and how to use it. Areas of importance and utility from RM-ODP are highlighted. Warnings in the use of RM-ODP are highlighted. Part Three provides a discussion of the principal architecture patterns of use, arranged by topic. Several of these patterns of use come from emerging work under the initiative of RM-ODP, as well as lessons learned from the practice of RM-ODP. These patterns of reasoning used by the architect are founded on the principals of RM-ODP, as discussed in Part Two of the book. Part Four concludes with relating RM-ODP to other architecture methods. It also provides emerging technologies to further the patterns of reasoning for use in architecting, and a set of architecting heuristics. The information contained in this book is organized in a manner that provides clear insight into the world of distributed software-intensive processing architecture for designers and developers who are familiar with information systems technology, but want to know more about how to build a good architecture. Starting with a tutorial about software architecture, and then a tutorial about the standard for software architecture, the reader need not be an expert in the area of international standards, RM-ODP, software architecture, or specific technologies. The book goes on to address the needs of the variety of readers for which it is intended. Each chapter in the book provides an overview of the subject of the chapter, as well as a summary. For those who wish a broad brush exposure to RM-ODP, the primers of Part One provide this, as well as the overviews and summaries in each chapter of interest. As each chapter progresses, in Parts Two and Three, more and more in-depth detail is provided. The readings of these chapters are aimed at those who wish to know the technical details of a topic. There are two case studies used throughout the book, at various levels of detail. The primary case study is a Hospital enterprise, based upon the author's experience with the medical profession. A secondary case study is an airline reservation system, also based upon the author's experience. These case studies are used to describe the concepts of RM-ODP, and to show how they might be used.
Architecting Mobile Solutions for the Enterprise

Your guide to planning and executing a complete mobile web strategy Revisit your approach to the mobile web--and deliver effective solutions that reach customers and clients on a variety of mobile devices. In this practical guide, web development luminary Dino Esposito shows you how to develop a solid mobile strategy for the enterprise, starting with an effective mobile website. You'll receive essential architectural and implementation guidance, as well as mobile-specific design patterns for bui...

Architecting Secure Software Systems

Traditionally, software engineers have defined security as a non-functional requirement. As such, all too often it is only considered as an afterthought, making software applications and services vulnerable to attacks. With the phenomenal growth in cybercrime, it has become imperative that security be an integral part of software engineering so that all software assets are protected and safe. Architecting Secure Software Systems defines how security should be incorporated into basic software eng...

Architecting Composite Applications and Services with TIBCO

The definitive guide to architecting reliable, effective composite applications and services with TIBCO products and technologies *Pragmatic, proven, comprehensive coverage: concepts, products, technologies, service design, service architecture patterns, federation, and much more *The official guide for TIBCO's new two-day education course covering these topics. *By Dr. Paul C. Brown, TIBCO's Principal Software Architect: the follow-up to his TIBCO Architecture Fundamentals. Today's complex co...

Architecting Complex-Event Processing Solutions with TIBCO(R)

*Complex-event processing is simple in principle but hard to do well in practice. This guide presents the principles and motivations for those new to the subject. More importantly, it details the entire thought-landscape of a complete implementation, using TIBCO products as the background. Well worth the read for anyone who is thinking of implementing a complex-event solution. Those who have already implemented one should read it as well, both for another perspective and for a view of the capabi...

Microsoft .NET - Architecting Applications for the Enterprise (Developer Reference)

Make the right architectural decisions up front and improve the quality and reliability of your results. Led by two enterprise programming experts, you'll learn how to apply the patterns and techniques that help control project complexity and make systems easier to build, support, and upgrade right from the start. Get pragmatic architectural guidance on how to: Build testability, maintainability, and security into your system early in the design Expose business logic through a service-ori...