A Framework for the Design of Large, Complex Distributed IT Systems

The Reference Model of Open Distributed Processing (RM-ODP) is an international standard that provides a solid basis for describing and building widely distributed systems and applications in a systematic way. It stresses the need to build these systems with evolution in mind by identifying the concerns of major stakeholders and then expressing the design as a series of linked viewpoints.

Although RM-ODP has been a standard for more than ten years, many practitioners are still unaware of it. Building Enterprise Systems with ODP: An Introduction to Open Distributed Processing offers a gentle pathway to the essential ideas that constitute ODP and shows how these ideas can be applied when designing and building challenging systems. It provides an accessible introduction to the design principles for software engineers and enterprise architects. The book also explains the benefits of using viewpoints to produce simpler and more flexible designs and how ODP can be applied to service engineering, open enterprise, and cloud computing.

The authors include guidelines for using the Unified Modeling Language™ (UML) notation and for structuring and writing system specifications. They elucidate how this fits into the model-driven engineering tool chain via approaches, such as Model-Driven Architecture® (MDA). They also demonstrate the power of RM-ODP for the design and organization of complex distributed IT systems in e-government, e-health, and energy and transportation industries.

All concepts and ideas in the book are illustrated through a single running example that describes the IT support needed by a medium-sized company as it grows and develops. Complete UML models and more are available at http://theodpbook.lcc.uma.es/

FEATURES

• Offers a concise, focused presentation of the essentials of RM-ODP and where it fits within today’s software processes
• Explains all the major concepts and mechanisms of the ODP framework
• Explores the latest developments in the ISO ODP standards
• Uses the widely adopted UML notation for modeling large open distributed systems using the ODP concepts
• Describes interoperability frameworks applicable to both government and industry sectors
• Presents a case study of a realistic IT system that illustrates the possibilities and advantages of the ODP approach
• Includes questions and practical exercises
• Provides complete UML models and more on a supporting website

CONTENTS

THE FRAMEWORK
What Is ODP About?
The ODP reference model
Viewpoints
Fundamental concepts
Useful building blocks
Service orientation
Human computer interaction
The right tools for the job

THE VIEWPOINTS
Enterprise Viewpoint
Designing with communities
Identifying roles

See reverse side for continuation of Contents and ordering information
CONTENTS continued...

Organizational structure
Roles and role filling
More than one community
Community behavior
Accountability and related concepts
Quality of service and other constraints
Identifying the system’s user interfaces
Writing enterprise specifications

Information Viewpoint
The primacy of information
The elements of the information language
Writing information specifications
Structure of the information specification
Relationship with other viewpoints

Computational Viewpoint
Designing with computational objects
Computational objects
Bindings
Interaction between computational objects
Environmental contracts and transparencies
Writing computational specifications
Relationship with other viewpoints

Engineering Viewpoint
What is the engineering viewpoint for?
Objects and distribution
Node architecture
Channel architecture
Common functions and processes
Writing engineering viewpoint specifications
Incorporating current technologies
Relationship with other viewpoints

Technology Viewpoint
Linking to the real world
The elements of the technology language

Relationship with other viewpoints
Correspondences—Joining It All Up
The need for correspondences
Different kinds of correspondence
Correspondences required by the ODP architecture
Anatomy of a correspondence specification
Taking a formal view
Examples of correspondences
Tool support for specifying correspondences

USING ODP
Conformance—Does It Do the Right Thing?
Compliance and conformance
A conformance community
Types of reference point
Conformance to viewpoint specifications
Claiming compliance or conformance

Transparencies—Hiding Common Problems
What is a transparency?
Types of transparency
Transparencies and viewpoints

Policies—Tracking Changing Requirements
Why do we need policies?
What is a policy?
Implementing policy

Federation—Talking to Strangers
How does interoperation work?
Interpreting and sharing information
The basis of interoperation
Engineering the federation
Federating type systems
Federating identity
Legacy systems
Interoperability or integration?

Using Existing Products
What does this product do for me?
Supplier and user views
Competing sets of viewpoints

System Evolution—Moving the Goalposts
Coping with change
The importance of tool support
Making changes to viewpoints
Avoiding synchronized transitions
Evolution of the enterprise
Version control

MOVING ON
Modelling Styles
The importance of formal models
What is a system?
Modelling open or closed worlds?
Capturing requirements
Expressing obligations
Expressing semantics

Sharp Tools
What should a tool do?
Model editors and analysis tools
Model-driven approaches
Model transformations
Languages for transformations
Viewpoints and transformations
More integration

A Broader View
Where to look next
Integration of other standards
Uses of ODP
Tools
Comparing enterprise architectures
Coda

Appendix A: The PhoneMob Specifications
Appendix B: Selected Exercises
Bibliography
Index