Preface

The Reference Model for Open Distributed Processing (the RM-ODP) is an international standard created by the standardization bodies ISO and ITU-T. It gives a solid basis for describing and building widely distributed systems and applications in a systematic way. Emphasis is placed on the need to build such systems with evolution in mind by identifying the concerns of major stakeholders and then expressing the design as a series of linked viewpoints representing these concerns. Each stakeholder can then develop an appropriate view of the system with a minimum of interference from the others.

Although ODP has been available as a standard for more than 10 years, standards are not easy bedtime reading. The ideas presented have an enthusiastic following, but, outside of it, many practitioners are still unaware of them. This book aims to provide a gentler pathway to the essential ideas that make up ODP and to show how they can be applied when designing and building real systems. It offers an accessible introduction to the design principles for software engineers and enterprise architects. In addition, it explains the benefits of using viewpoints to produce simpler and more flexible designs. It is not limited to any single tool or design method, but concentrates on the key choices that make an architectural design robust, flexible and long lived. The book also shows the power of enterprise architecture for the design and organization of complex distributed IT systems.

The book has been prompted by the recent revision of the standard, during which ISO has incorporated experience from the application of ODP to many different domains and has taken account of new technologies and fashions. We cover the most up to the minute developments in the ODP standards, including the recent updates to the ODP reference model and the ODP enterprise language. The book provides fresh insights into the design of enterprise systems. Another reason for producing this book is to mark the publication of the ISO/IEC 19793 standard (known as UML4ODP), which uses the Unified Modelling Language (UML) notation to provide a familiar and accessible way of expressing ODP designs; it does this by defining a standardized UML profile. The book also provides guidelines for using the UML notation for structuring and writing system specifications and for fitting such specifications into the Model-Driven Engineering toolchain. This gives users of the ODP ideas a simpler way of expressing them with existing design and modelling tools.

Finally, there is an ongoing interest in using ODP when addressing new standardization approaches and interoperability frameworks such as those in

e-government, e-health, and the energy and transportation industries. The book shows how the RM-ODP ideas can be applied to modern movements such as service engineering, cloud computing and the creation of the open enterprise.

This is the first book to introduce the ODP material in language that is common to software engineers and software architects. It offers a short, concise and focused presentation of the essentials of RM-ODP and shows where it fits within today's software processes. The book describes all the major concepts and mechanisms of the ODP framework, explains how to use them in a practical way for the specification of large open distributed systems, and presents the basic notation used for creating the specifications. It follows the standards faithfully, but provides extra information on the thinking behind them, and on how they should be interpreted. The reader can get acquainted with the best design concepts and practices, which are essential to anyone who designs large software applications professionally.

The Roadmap

The book is targeted at a number of different audiences. It has been written to be attractive both to the technical experts working on system architecture and to a much broader audience working on realizing such systems.

The book is divided into four parts, each having a different focus and each exploring progressively more detail of the different concepts and their use. There are also two appendices. The four parts provide

- An extended executive summary introducing the basic structuring ideas of ODP, particularly the central idea of there being a set of viewpoints.
- A more detailed explanation of the five ODP viewpoints and of the correspondences between them. Reading this part gives an idea of the style and use of the main elements of the ODP architecture.
- An explanation of the way these concepts are used to solve a number of the common problems met in the development and evolution of distributed systems. This part will help the reader to understand how use of the structure results in more flexible and adaptable systems.
- A discussion of some of the subtler ideas underlying this kind of system modelling and the new requirements they place on the supporting tools. This part answers some of the immediate questions people often want to ask about why the framework was defined in the way it is.

The first appendix brings together all the example model fragments used in the book to provide a single overview of a simple ODP-based design. Space

limitations mean that even this needs to be selective, and a fuller version is available from the authors' website (see `http://theodpbook.lcc.uma.es/`).

The second appendix presents some questions and scenarios that can be used to support teaching and training using this book. In addition to providing information for practicing professionals, we expect the book to provide a resource for graduate students and researchers who want to understand the main problems and principles involved in the design of large software systems; the book can also be used in Masters or Doctorate courses for teaching the concepts and design principles of ODP and for preparing students to research new problems in this area.

The whole structure is unified by the use of a single running example describing the IT support needed by a medium-sized company as it grows and develops. One of the problems in understanding a system's architecture is in seeing just how it helps the day-to-day activities of the system builders. Abstract structures in their purest form can seem dry and remote, so we have tried to relate them to the problems developers face by including a series of short vignettes illustrating why the various aspects of the architecture are needed. These fragments support, but do not form an essential part of, the main exposition. The individual chapters generally start with one of these fragments to give an informal introduction to the problem to be solved.

The book is targeted at three groups of readers. It is primarily intended for enterprise architects and software engineers who want to understand the concepts, mechanisms and problems involved in the design of complex enterprise systems and to use this knowledge in establishing a tool-based approach to documenting, evolving and testing systems. Readers will become more aware of the issues and options available for designing within a strong architectural framework. They can apply the ideas in general terms, or study the full detail further in the standards or in one of the reference books based on them.

A second target audience includes IT project managers and CIOs, who will be able to understand the possibilities of the RM-ODP framework and a viewpoint-based design approach, and the potential benefits that the adoption of this approach can bring to their companies and organizations. There are a growing number of large, multi-organizational information systems projects, for example in the aerospace and healthcare areas, and the designers involved are seeking urgently for a systematic architectural approach. ODP provides such an approach.

A third group, CEOs and business architects, can get an overall idea of this design approach and be able to evaluate the advantages of the use of a mature set of ISO and the ITU-T standards within their organizations, and also for interoperating with the IT systems of their customers, providers, financial services, and so on. The use of international standards is now essential to achieve (and to guarantee) the level of interoperability required in these large and complex IT systems with hundreds of customers, providers and developers, which need to exchange data and services with other IT systems in a seamless way.

Historical Background

The requirements for ODP and its reference model can be traced back a long way, from the earlier work on Open Systems Interconnection. It became clear to experts working on application protocols that the prevalent focus on peer-to-peer communications was not enough and that distributed systems design needed to take a more holistic approach, starting from the structure of the organizations involved. It was necessary to begin with a thorough understanding of the enterprise before proposing any technical solutions.

At the same time, there was considerable work in progress on the idea of middleware supporting a uniform distribution platform, leading, for example, to the ANSA architecture [72] and to early Object Management Group (OMG) specifications. The work on ODP started by harvesting the current research ideas available at that time and then used them to construct a vendor-neutral architecture. This principle of maintaining a broadly applicable framework by using the best current thinking has continued to be the basis for work on ODP.

Currently RM-ODP is being maintained and developed by an ISO/IEC Working Group (JTC1/SC7/WG19: *Techniques for the Specification of IT Systems*). This group works in close cooperation with ITU-T on the joint publication of a series of standards, and also maintains strong collaborative links with other international bodies, such as the OMG. These links help to provide the intelligence for continuously updating and improving the framework as new technologies and paradigms emerge, and for maintaining the consistency of a broad range of specifications and standards.

Conventions Used

Much of the explanation of architectural elements is about the way different concepts are represented and what artefacts might result from their application. It is easy to lose track of which use falls into which category, so we introduce some graphical conventions to help sort things out. Throughout the book, we will use the following typographical conventions:

- A ***bold italic*** font highlights a word that is an ODP concept.
- A sans serif font indicates that an element is from a UML model.
- A `typewriter` font flags an item as being a concrete instance.

Trademarks and Copyright

Acknowledgements

The ODP standards are the result of the efforts of a great many experts working within the standards process over many years. There is not room here to give credit to all of these people, but we should like, in particular, to acknowledge the contribution of the ISO conveners and document editors in leading the effort, particularly Joost J. van Griethuysen, Eng Chew, Bryan Wood, Jean Bérubé, Fausto Caneschi, Jean-Bernard Stefani, Andrew Herbert, Richard Sinnott, Mirion Bearman, Pramila Daryani, Peter Furniss, Lea Kutvonen, Laurent LeBoucher, Joaquin Miller, Kerry Raymond, Gerd Schürmann, and Sandy Tyndale-Biscoe. Arve Meisingset played a pivotal role in coordinating our link with the ITU-T, and Tom Rutt provided vital liaisons links with the OMG.

A number of people have read drafts of this book and provided us with very useful suggestions; in particular, our thanks go to Andy Bond, Fred Cummins, Andrew Herbert, Jishnu Mukerji, Bruno Traverson and Bryan Wood.

The authors would also like to thank No Magic, Inc. for making their UML tools available to facilitate the production of this book, and José Raúl

Romero and Juan Ignacio Jaen for building and maintaining the ODP plug-in for MagicDraw, and for maintaining the RM-ODP website.

Finally, we would like to thank all those involved at CRC Press, particularly Randi Cohen, Amber Donley and Karen Simon, for their expert help and advice throughout the publication process.

Peter F. Linington
Zoran Milosevic
Akira Tanaka
Antonio Vallecillo