BUILDING AN OPEN SYSTEM

Jacob Slonim
Abraham Schonbach
Michael A. Bauer
Lachlan J. MacRae
Keith A. Thomas

BUILDING AN OPEN SYSTEM

Jacob Slonim
Abraham Schonbach
Michael A. Bauer
Lachlan J. MacRae
Keith A. Thomas

VNR VAN NOSTRAND REINHOLD COMPANY
New York

Library of Congress Catalog Card Number 86-26743
ISBN 0-442-28068-8

Printed in the United States of America

Van Nostrand Reinhold Company Inc.
115 Fifth Avenue
New York, New York 10003

Van Nostrand Reinhold Company Limited
Molly Millars Lane
Wokingham, Berkshire RG11 2PY, England

Van Nostrand Reinhold
480 La Trobe Street
Melbourne, Victoria 3000, Australia

Macmillan of Canada
Division of Canada Publishing Corporation
164 Commander Boulevard
Agincourt, Ontario M1S 3C7, Canada

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Library of Congress Cataloging-in-Publication Data

Building an Open System

Bibliography; p.
Includes index.
1. Electronic data processing—Distributed processing.
I. Slonim, Jacob, 1945- .
QA76.9.D5064 1987 004'.36 86-26743
ISBN 0-442-28068-8

Preface

A great many books and articles have been written in recent years on open systems, but most of these concentrate on the communications issues, and often tend to leave the impression that once the communications problems are solved, the world of open systems will have arrived. This is by no means the case. One of the aims of this book is to correct that impression by describing the ways in which *all* aspects of the computing environment, in both hardware and software, can promote or impede the realization of the open systems ideal. It is argued, for example, that while new hardware and software are essential to the development of open systems, the new environments must also be able to accommodate the hardward and software, particularly the latter, we use today. This book presents rather more comprehensively than usual the technology of open systems, and attempts to predict the development of that technology in a historical context in which technical, economic, and political problems are intertwined.

JACOB SLONIM
ABRAHAM SCHONBACH
MICHAEL A. BAUER
LACHLAN J. MACRAE
KEITH A. THOMAS

Abbreviations and Acronyms

2PC	Two Phase Commit
2PL	Two Phase Locking
ACID	Atomic, Consistent, Isolated, Durable
AI	Artificial Intelligence
ALPS	Automated Language Processing System
ANSI	American National Standards Institute
ARPANET	Advanced Research Projects Agency Network
BNF	Backus-Naur Form
BSC	Binary Synchronous Communications
CAD/CAM	Computer Aided Design/Computer Aided Manufacturing
CASE	Common Application Service Elements
CATV	Cable TV
CCIS	Common Channel Interoffice Signaling
CCITT	Consultative Committee for International Telephone & Telegraph
CCS	Common Channel Signaling
CICS	Customer Information Control System
CISC	Complex Instruction Set Computer
CMEA	Council for Mutual Economic Assistance
CODASYL	Conference on Data System Languages
COS	Corporation for Open Systems
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CSP	Cooperating Sequential Processes
DARPA	Defense Advanced Research Projects Agency
DB2	Database2 (IBM Relational Database Management System)
DBMS	Database Management System
DCE	Data Communication Equipment
DDBMS	Distributed Database Management System
DDL	Data Description Language
DDS	Dataphone Digital Service
DDTS	Distributed Database Testbed System
DEC	Digital Equipment Corp.
DML	Data Manipulation Language
DOD	Department of Defense

DSA	Directory System Agent
DTE	Data Terminal Equipment
DUA	Directory User Agent
ECC	European Economic Community
ECMA	European Computer Manufacturers Association
EIA	Electronic Industries Association
ER	Entity Relationship (Model)
ESPRIT	European Strategic Program for Research
FAX	Facsimile
FCC	Federal Communications Commission
FDM	Frequency Division Multiplexing
GA	Gate Array
Gbps	Gigabits per Second
HP	Hewlett Packard
IBM	International Business Machines
IC	Integrated Circuits
ICOT	Institute for New Generation Computer Technology (Japan)
IDP	Internet Datagram Protocol
IEEE	Institute of Electrical and Electronic Engineers
IMS	Information Management System
IRDS	Information Resource Directory System
ISO/OSI	International Standards Organization/Open System Interconnection
ISDN	Integrated Services Digital Network
KBS	Knowledge Based System
KR	Knowledge representation
Kbps	Kilobits per Second
LAN	Local Area Network
LEONARDO	Low-cost Exploration Offered by the Network Approach to Requirements & Design
LU	Logical Unit
MAILA	Mail Agent
MAP	Manufacturing Automation Protocol
MCC	Microelectronics & Computer Technology Corp.
MF	Mail Forwarder
MIPS	Microprocessors without Interlocked Pipe Stages (Stanford U)
MIPS	Millions of Instructions per Second
MIS	Manager Information System
MISC	Microelectronics & Information Sciences Center (Minneapolis)
MITI	Ministry of International Trade & Industry (Japan)

MNC	Microelectronics of North Carolina
MS	Mail Server
MVS	Multiprogramming with Virtual Storage
Mbps	Megabits per Second
NBS	National Bureau of Standards
NCR	National Cash Register
NIL	Network Implementation Language
NLI	National Language Interfaces
NT1	Network Terminator Type 1
NT2	Network Terminator Type 2
OS	Operating System
OSIE	Open System International Environment Optimization
PABX	Private Automated Branch Exchange
PBS	Private Branch Exchange
PC	Personal Computer
PCI	Protocol Control Information
PCL	Process Control Language
PCM	Pulse Code Modulation
PDL	Program Development Language
PDU	Protocol Data Unit
PLA	Programming Logic Array
PLANET	Programming Language for Networks
PR	Procedural Representation
PSS	Packet Switching Service
PTT	Postal Telephone and Telegraph Authority
RAM	Random Access Memory
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
SDU	Service Data Unit
SEISMU	Software Engineering Institute at Carnegie Mellon University
SFP	Standard Floor Plan
SISAL	Streams & Interaction in a Single Associated Language
SNA	System Network Architecture
SNFS	Sun Network File System
SPC	Software Productivity Consortium (Washington, D.C.)
SQL	Sequential Query Language
SRC	Semiconductor Research Corp.
SWN	System Wide Name
SC	Standard Cell
TAB	Tape Automated Bonding
TDM	Time Division Multiplexing
TC97/SC16	Technical Committee 97, Subcommittee 16

UDS	Universal Directory System
VLSI	Very Large System Integration
WAN	Wide Area Network
XCON	Expert Configurer
XDFS	Xerox Distributed File System for Research in Information Technology

Contents

Chapter 1

Introduction

1.1 DISTRIBUTED SYSTEMS AND OPEN SYSTEMS

In the early 1970s, all but the largest organizations depended on centralized computer centers, through which information flowed in carefully controlled batches, or to which some users were connected with interactive access to very limited processing functions. Networks of computers had begun to appear in some large organizations, but, with the exception of experimental projects such as ARPANET, they reflected the established practices of compartmentalization, hierarchical control, and batch processing. In the mid-1980s, the smallest organizational components—even individuals—can afford interactive computing resources dedicated to their own needs. As well, the operation of most organizations has become more event-driven and dependent on immediate communication and access to information: less oriented to batch processing and more to interactive. This applies both within and between organizations. Decentralized processing has emerged as both affordable and necessary.

Even with this explosion of cheap computing resources, and the changing pattern of organizational behavior, the MIS manager or the systems designer has not escaped the necessity of providing coherent, secure, and reliable systems that serve the needs of the whole organization, not just its individual parts. Activities must be coordinated according to organization-wide rules, and information must be collected, collated, and channeled to appropriate operating units without continual human intervention. In short, information systems that exhibit both the stability of the centralized approach and the flexibility of the decentralized approach must be constructed out of a mosaic of virtually independent processing systems. Two related concepts are involved in a solution to this problem: open systems and distributed systems.

In the broadest context, not restricted to computer systems, an open system is characterized by flows of matter, energy, or information into and out of the system across its natural boundaries. In the context of computer systems, open systems are those characterized by similar flows of data across boundaries between autonomous systems. This entails not only mechanisms to facilitate the flow of data, but also a positive intent on the part of the system's administrators to encourage this sort of traffic.

The term *open system* has been most widely applied to computer systems in connection with the International Standards Organization's Reference Model for Open System Interconnection (ISO/OSI), an internationally accepted template for computer communications standards. An open system in the context of ISO/OSI entails the sharing of control among the members of a system, and direct interaction among such members.

It is possible to create a data-flow between systems via off-line bulk media such as magnetic tape, but control of such transfers is divided into two clearly separated domains: first one system produces a data batch according to its own schedule, then another system processes that batch according to its own schedule. This mode of interaction might at first seem to constitute an open system. Lacking here, however, are the shared control and direct interaction required in transfer by means of two-way continuous channels.

Therefore, in this context, "open system" will refer to a computer system organized and operated in such a way that it can be easily connected to other independently organized and operated computer systems through telecommunication links in order to exchange data on an as-needed basis.

An important aspect of open systems is that specific policies must be instituted and the deliberate actions must be taken in order to enable interconnection and interchange. This cannot happen without the active and positive intervention of a system's administrators. While one can imagine a time when data exchange and intersystem transactions might be almost automatic, perhaps using symbolic directory and translation services embedded in the telecommunications network itself, such capabilities cannot be realized in the present or foreseeable state of the art. Therefore, the implementation of even the simplest degree of open system requires intense cooperation on the part of the administrators of the systems involved.

It should be obvious that the necessary flexibility, especially to cope with growth and technological change, requires that the interfaces between systems be well-defined and widely available, that is, adhere to standards. To a large extent the practicability of open systems is tied to the existence of industry-wide national and international standards; therefore, this book will concentrate heavily on applicable existing and evolving standards.

Another important aspect of open systems is that their components are treated as equals, in what ISO calls peer-to-peer relationships. There are no master-slave relationships, no hierarchy or central control, though there may be distinguished processors within a network that provide unique services. This condition has implications for the way in which coordinated activities can be implemented in open systems.

The open system supports decentralized processing and provides the flexibility to adapt to change, but what supplies the stability for overall control and coordination? There is, at least in a broad sense, a need to take a tightly coupled, centralized information system and "divide it up" over a number of processing sites in such a way that activities can be coordinated and global rules applied, but at the same time that each site be allowed as much local autonomy as possible. This is the concept of a *distributed system.*

Distributed systems rely on the sharing of control and the interactive exchange of information in order to effect overall coordination, much like the communications capabilities required by open systems. In fact, any implementation of a communication protocol that includes the exchange of control or state information as well as data is a distributed system, if only with respect to communications processing. It is possible to conceive distributed systems that are not open systems: systems that are not organized for easy interconnection and data exchange beyond their immediate partners, that utilize non-standard communication techniques, or that are centrally or hierarchically controlled. Such distributed systems support decentralized processing only in a limited way, lacking flexibility and local autonomy. While complete local autonomy—absolute equality among peer systems—is not possible in distributed systems, it can be approached through algorithms built on cooperation and dynamically shared control, exactly as the interfaces within an open system are defined.

The concepts of an open system and a distributed system are separate but strongly related. They are both important in implementing decentralized processing in a contemporary organization. There are three principal problem areas in which both open system and distributed system features are useful:

1. *Data sharing or exchange among separate processing centers.* Open system features such as application protocols are required; however, where parts of the same organization are involved, distributed system features may also be required to enforce stronger rules for coordinating transactions.
2. *Decentralizing from a centralized processing center.* One of the classic reasons for decentralization is to reduce contention, response time, and system overhead costs associated with an overloaded central facility, and to reduce system communication costs, by providing separate processing facilities located closer to their users. Distributed system features are clearly required, but open system features are desirable for growth and flexibility.
3. *Coordinating the activities of separate processing centers.* Where it is necessary to bring some coherence to independent processing

centers (e.g., linking individual workstations into an organization-wide network), open system features are required to effect the flow of data, and distributed system features are required to coordinate activities.

Open systems in support of distributed systems, and distributed systems in the context of open systems, constitute the framework in which practical problems must be approached. Therefore, this book will focus on both and on the relationship between them. Since at present open system features are defined mainly with respect to communication interfaces, the chapters dealing with communication and network services will concentrate on open system issues, whereas the chapters on overall design, OS, and database will concentrate on distributed system issues. Throughout, discussions concerning the trade-offs between local autonomy and overall coordination will examine the interplay between the two concepts.

1.2 THE BOOK: PURPOSE AND STRUCTURE

The past few years have produced a number of books on distributed information systems, devoted for the most part to communications issues and to distributed databases. Such an approach is quite reasonable, since both these areas are critical to the understanding of distributed systems. However, the result of this bias is that many other issues which are of importance to practitioners have been addressed only superficially. We know of no work which provides a "holistic" view of distributed systems, still less of open systems. Indeed, there has as yet been little detailed discussion of the notion of an open system and its implications. Nor has there been much direct analysis of the real problems encountered in developing distributed systems in the context of existing systems hardware and software.

What is required, in the authors' opinion, is a global perspective of distributed systems and open systems, encompassing relevant aspects of languages, operating systems, communications, and other tools. The present work hopes to supply such a view.

The argument of the book centers on two key issues:

1. The concept of an open system, and
2. The fact that any practical approach to distributed systems must be built upon and from existing systems.

If there is a single principal theme, it is that practical distributed systems, and thus practical open systems, must be developed in an evolutionary manner from current systems rather than in a revolutionary way: that is to say, we cannot simply discard existing systems.

After this brief introduction we proceed to an overview of those technological advances and trends which form the basis of open systems, present and future. *Chapter 2* reviews trends in a number of areas which have a direct impact on the design and development of open systems, offering brief introductions to many topics which are later treated in detail. There are sections on Hardware, Software, Communications, and Artificial Intelligence.

Hardware. After a general discussion of computer architectures, the section proceeds to specific techniques and devices which are influencing current developments in distributed systems. Among these are the automated layout of integrated circuits, reduced-instruction-set architectures, and microcoding. Apart from the RISC machine, two new types of computer are given special attention: the "multicomputer" and the 32-bit microprocessor.

Software. After a discussion of the general conditions of software in an open, distributed environment, the section focuses on three critical areas. First, distributed database management systems—the heart of any distributed information system. The next topic is Ada, the high-level programming language which is rapidly assuming the status of a de facto standard under the sponsorship of the U.S. Department of Defence. The third issue is the operating system: What qualities are required of an operating system in a distributed environment? The emphasis is on UNIX, which is of course also becoming a de facto industry standard: How adaptable is UNIX to the open environment, and what changes must be made? The section concludes with a survey of commercial and experimental distributed database management systems.

Communications. Rather than attempt a survey of communications tools (for that see Chapter 6), this section concentrates on that single development which holds the greatest promise for the future, the integrated services digital network (ISDN). The ISDN is discussed at greater length later in the book; the overview here is intended to introduce the reader to the basic issues: what it is, what it has to offer, its drawbacks, and its likely course of evolution.

Artificial Intelligence. Since this final section deals with a topic with which many readers are only marginally familiar, the aim is to define the field in general terms, to describe the more important component disciplines of AI, and to show how these disciplines may influence the development of open systems. The section, and the chapter, concludes with a summary of the more important AI research projects.

Chapter 3, on the architecture of open systems, moves from the general to the particular, to the specific techniques whereby distributed and open systems are implemented. The first section discusses the logical architecture, the various ways in which it is convenient to view the structure of an open system. The discussion then concentrates on one of the layers of that structure, the operating system: specifically its role in communications, file management, naming, and database support. The problems of heterogeneous operating systems in a distributed system occupy the next section, together with various prospective solutions: limited-set, portable, and virtual operating systems. The chapter concludes with a more general account of the impact of technological advances on open system architectures: the effect of specific types of new device, and above all the need to devise architectures which can accommodate technological change.

Chapter 4, on directories and dictionaries, is an account of the basic means of locating and identifying all the various elements of an open information system. While such tools are important in any kind of information system, in a distributed system they are crucial, and must be very powerful indeed. To avoid ambiguities, the chapter begins with a set of definitions. The second section describes data dictionaries: their uses and benefits, and the techniques by which they are implemented. Naming techniques as they pertain to data dictionaries are the subject of 4.3, which concludes with descriptions of the naming facilities associated with a number of distributed systems.

Then there is a description of the Universal Directory System developed at Stanford, presented in some detail because it embodies a great number of features desirable in open system directories. Naming systems as they are used in open system directories are discussed in section 4.5, and the following two sections deal with two specific types, machine-oriented and human-oriented naming. Section 4.8 is a brief treatment of the problems of network directories, and 4.10 deals with the management of catalogues in distributed databases. The concluding sections concern the conventions for naming and addressing used in the context of the ISO Reference Model for Open Systems Interconnection (ISO/OSI), and certain implementation issues.

Chapter 5, on the design and implementation of open systems, is essentially a discussion of the principles of software engineering in relation to distributed and open systems. It therefore contains accounts in this special context of each of the phases (beyond requirements) of the software life-cycle: specification, design, implementation, integration and testing, and maintenance. The chapter centers on its most extensive section, which first enumerates what is required of a high-level language in a distributed environment, and then describes and evaluates a number of languages (e.g., Ada, Modula II, Concurrent "C") in the light of those requirements.

Chapter 6 is a survey of the communications tools which are or soon will be available for use in distributed and open systems. The framework of this survey is the ISO/OSI Reference Model. The chapter treats four main subject areas. The first of these is the notion of open systems in communications. An explanation of the open systems concept is followed by a discussion of the ways standards emerge, and then by a description of the ISO/OSI standards themselves.

The second subject is the current state of the art in communications technology. Rather than describing the entire field, we highlight the more significant developments and trends in local area networking, long-haul communications, and network interconnection. The third subject area contains some important practical aspects of communications, including security, performance, and network management. The chapter concludes with an extensive treatment of the integrated services digital network: its benefits, costs, and problems, and what it will offer in both the shorter and longer terms.

Chapter 7, on distributed databases, resembles Chapter 4 in that it redefines terms and techniques well understood in relation to centralized databases for use in distributed systems. A set of precise definitions, presented at the outset, allows quick movement to a discussion of issues pertaining to the DBMS architecture, data models, and conceptual design, all in a distributed environment. Two issues which are of particular importance in a distributed database system, query processing and transaction processing, are given special treatment. Other problems (physical design, administration, definition and description, and communications) are examined in a more cursory manner—most of these are issues which have been treated in other chapters, seen from a DBMS point-of-view. The chapter concludes with a discussion of suitable applications for distributed DBMSs, and a listing of commercial distributed database systems.

Our final chapter, "Future Developments" is concerned not so much with the technology of open system as with the way in which open systems will be used in the near future to support various cooperative activities. Our first example is MCC (Microelectronics and Computer Technology Corporation), which is a consortium formed by a number of major American manufacturers of computing equipment. The aim of MCC is to conduct basic research on behalf of all these companies: all will contribute to the input of ideas, and all will share in the output. To support the very ambitious projects of MCC a powerful, distributed, and above all *open* information system is required—it must link and render into a single system all the disparate facilities of the member companies.

The same is true of our second set of examples, national and international research and development projects such as the French "National