A James Martin BOOK

# SYSTEMS APPLICATION ARCHITECTURE

## Common Communications Support: Distributed Applications

**JAMES MARTIN**

with

Kathleen Kavanagh Chapman / Joe Leben

PRENTICE HALL
Englewood Cliffs, New Jersey 07632

*Library of Congress Cataloging-in Publication Data*

Martin, James

Systems application architecture : common communications support : distributed applications / James Martin with Kathleen Kavanagh Chapman, Joe Leben.

p. cm.

"The James Martin books."

Includes bibliographical references and index.

ISBN 0-13-785908-2

1. IBM Systems Application Architecture. 2. Electronic data processing—Distributed processing. I. Chapman, Kathleen Kavanagh. II. Leben, Joe. III. Title.

QA76.9.A73M373 1992

004'.36—dc20 91-31181

CIP

Editorial/production supervision: Kathryn Gollin Marshak
Jacket design: Bruce Kenselaar
Prepress buyer: Mary Elizabeth McCartney
Manufacturing buyer: Susan Brunke
Acquisitions editor: Paul W. Becker

Published by Prentice-Hall, Inc.
A Division Simon & Schuster
Englewood Cliffs, New Jersey 07632

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-785908-2

ISBN 0-13-785908-2

90000>

9 780137 859085

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

# SYSTEMS APPLICATION ARCHITECTURE: COMMON COMMUNICATIONS SUPPORT

## Distributed Applications

A James Martin BOOK

## THE JAMES MARTIN BOOKS

currently available from Prentice Hall

- Application Development Without Programmers
- Building Expert Systems
- Communications Satellite Systems
- Computer Data-Base Organization, Second Edition
- Computer Networks and Distributed Processing: Software, Techniques, and Architecture
- Data Communication Technology
- DB2: Concepts, Design, and Programming
- Design and Strategy of Distributed Data Processing
- An End User's Guide to Data Base
- Fourth-Generation Languages, Volume I: Principles
- Fourth-Generation Languages, Volume II: Representative 4GLs
- Fourth-Generation Languages, Volume III: 4GLs from IBM
- Future Developments in Telecommunications, Second Edition
- Hyperdocuments and How to Create Them
- IBM Office Systems: Architectures and Implementations
- IDMS/R: Concepts, Design, and Programming
- Information Engineering, Book I: Introduction and Principles
- Information Engineering, Book II: Planning and Analysis
- Information Engineering, Book III: Design and Construction
- An Information Systems Manifesto
- Local Area Networks: Architectures and Implementations
- Managing the Data-Base Environment
- Principles of Data-Base Management
- Principles of Data Communication
- Recommended Diagramming Standards for Analysts and Programmers
- SNA: IBM's Networking Solution
- Strategic Information Planning Methodologies, Second Edition
- System Design from Provably Correct Constructs
- Systems Analysis for Data Transmission
- Systems Application Architecture: Common User Access
- Systems Application Architecture: Common Communications Support: Distributed Applications
- Systems Application Architecture: Common Communications Support: Network Infrastructure
- Systems Application Architecture: Common Programming Interface
- Technology's Crucible
- Telecommunications and the Computer, Third Edition
- Telematic Society: A Challenge for Tomorrow
- VSAM: Access Method Services and Programming Techniques

*with Carma McClure*

- Action Diagrams: Clearly Structured Specifications, Programs, and Procedures, Second Edition
- Diagramming Techniques for Analysts and Programmers
- Software Maintenance: The Problem and Its Solutions
- Structured Techniques: The Basis for CASE, Revised Edition

TO CORINTHIA
*—JM*

TO JOHN AND MY PARENTS
*—KKC*

TO ERIN, BECAUSE IT'S A TOP-DOWN BOOK
*—JFL*

# PREFACE

IBM's intent with *Systems Application Architecture* is to provide a standard set of interfaces to computing for both application developers and end users. These interfaces are intended to provide a framework for developing consistent computing applications that operate in all of IBM's major computing environments.

One of IBM's motivations for developing SAA is to address the problems caused by the lack of commonality between the different computing environments that it supports. Although many of the same functions are provided in each computing environment, different programs must generally be used to provide these functions. The programs used in the various environments are often dissimilar in how users access their functions and in the results they produce.

SAA provides a strategic direction for the use of IBM computing equipment and software and has the potential to become a standard that defines a universal computing environment. In this environment, applications can be developed without regard to the underlying hardware or operating system software.

SAA defines three major interfaces. The *Common User Access* (CUA) interface specifies how an application developer creates an SAA-compliant user interface. The *Common Programming Interface* (CPI) specifies a standard set of languages and a standard set of services that can be used for application development. *Common Communications Support* (CCS) defines both high-level services and a low-level networking infrastructure that allow standarized communication among devices, application programs, computing systems, and computer networks.

A major goal of CCS is to provide facilities that allow application developers to construct distributed applications that use computer networks for communication. In such applications, the computer processing is distributed among two or more computing systems, and access may be provided to resources that may also be distributed among several processors.

CCS includes support for a variety of high-level data object formats, data stream structures, and application services. These allow application developers to interconnect IBM and non-IBM systems to create distributed applications and to allow computing systems to have access to distributed resources. CCS also defines lower-level services that must be provided by the underlying networking infrastructure to support the transmission of information through a computer network.

The architects of Common Communications Support recognized that two computer network architectures are important in the IBM networking environment. The first of these is IBM's own *Systems Network Architecture* (SNA) that was first described in the mid-1970s and has been evolving ever since. Most of today's large IBM installations use networking equipment and software conforming to the SNA formats and protocols. Of increasing importance in today's communications environment are the international standards that fall under the umbrella of the *Reference Model of Open Systems Interconnection,* or OSI model for short, developed by the International Organization for Standardization (ISO). The applicable ISO standards together make up the OSI architecture. CCS includes important elements of both the SNA and the OSI architectures.

This book takes a top-down approach and examines in detail the high-level data object formats, data stream structures, and high-level application services that CCS supports for both the SNA and OSI environments. These together provide the services that application developers use in the CCS environment to build distributed information processing applications. A more technically oriented companion volume to this one takes a bottom-up approach and describes in detail the data link controls, network services, and session service that are used to connect an SAA processor to computer networks that use either the SNA or the OSI protocols. Other books in the James Martin SAA series describe the Common User Access and Common Programming Interface components of SAA.

## PLAN OF THE BOOK

This book begins with a chapter that introduces the structure and components of Systems Application Architecture, showing how SAA Common Communications Support relates to the other SAA architectural components.

Part I then describes the characteristics of distributed systems. Chapters in Part I discuss the important topic of information systems standardization, provide brief introductions to both the SNA and OSI architectures, introduce the six major elements of CCS, and describe how CCS services can be used to design and implement a distributed computing environment.

Part II concentrates on the transmission objects that make up one of the six elements of CCS. The chapters in Part II describe the characteristics of office information systems and introduce IBM's Information Interchange Architecture (IIA) from which many of the components of CCS were derived. Chapters in Part II also describe the various architectures that are included in CCS for defining the structure and content of transmission objects, including the Presentation Text Object Content Architecture (PTOCA), Graphics Object Content Architecture (GOCA), Image Object Content Architecture (IOCA), Formatted Data Object Content Architecture (FDOCA), and the Font Object Content Architecture (FOCA). Part II ends with a chapter on revisable documents, for which support is not yet included in CCS.

Part III describes architectures that are included in CCS for defining the structure and content of the data streams that carry the transmission objects discussed in Part II. The chapters in Part III describe the Mixed Object Document Content Architecture (MO:DCA) data stream, the Intelligent Printer Data Stream (IPDS), the 3270 Data Stream, and the Revisable Form Text Document Content Architecture data stream (RFT:DCA) data stream. Part III ends with a chapter on the Character Data Representation Architecture (CDRA) that is concerned with maintaining the integrity of textual data in the distributed computing environment.

Part IV describes a number of IBM-defined architectures that discuss how application services are supplied in a network conforming to the SNA architecture. The chapters in Part IV describe the Document Interchange Architecture (DIA), SNA Distribution Services (SNA/DS), the Distributed Data Management (DDM) architecture, the Distributed Relational Database Architecture (DRDA), and SNA Management Services (SNA/MS).

Part V describes a number of international standards that discuss how application services are supplied in the Open Systems Interconnection (OSI) environment. The chapters in Part V describe the Association Control Service Element (ACSE), CCITT Recommendation X.400, File Transfer, Access, and Management (FTAM), and the IBM OSI/File Services family of OSI software products that implement the FTAM standard.

Part VI briefly introduces the elements of CCS that define the underlying SNA and OSI networking infrastructure. The chapters in Part V introduce SNA and OSI session services, SNA and OSI network services, and data link controls.

Part VII includes chapters that introduce the application programming interfaces that are supported in CCS for requesting communication services in both the SNA and OSI environments. The Common Programming Interface for Communications (CPI-C) is the API that CCS defines for accessing communication services in the SNA environment. The OSI/Communications Subsystem family of OSI software products defines an API that can be used for accessing communications services in the OSI environment.

## ACKNOWLEDGMENTS

The authors wish to thank John Gridley of IBM, who provided us with much helpful information and got us in touch with the right people at IBM who were able to answer our many questions. We would also like to thank our many friends in the Networks and Communications group of the Digital Equipment Corporation who helped us to get the OSI concepts straight.

*James Martin*
*Kathleen Kavanagh Chapman*
*Joe Leben*

# LIST OF ACRONYMS

**ACSE.** Association Control Service Element.

**AET.** Application-entity-title.

**AM.** Amendment.

**ANSI.** American National Standards Institute.

**APDU.** Application-protocol-data-unit.

**API.** Application-process-invocation.

**API.** Application programming interface.

**APPN.** Advanced Peer-to-Peer Networking.

**ASC.** Accredited Standards Committee.

**ASE.** Application-service-element.

**ASN.** Abstract syntax notation.

**ASN.1.** Abstract Syntax Notation One.

**BER.** Basic Encoding Rules.

**BLU.** Basic Link Unit.

**CCITT.** International Telegraph and Telephone Consultative Committee.

**CCS.** Common Communications Support.

**CD.** Committee draft.

**CDAM.** Committee Draft Amendment.

**CDRA.** Character Data Representation Architecture.

**CLNS.** Connectionless-mode Network Service.

**CONS.** Connection-mode Network Service.

**CPI.** Common Programming Interface.

**CSMA/CD.** Carrier Sense Multiple Access with Collision Detection.

**CUA.** Common User Access.

**DAD.** Draft Addendum.

**DAM.** Draft Amendment.

**DCE.** Data circuit-terminating equipment.

**DDM.** Distributed Data Management.

**DIA.** Document Interchange Architecture.

**DIS.** Draft International Standard.

**DLPDU.** Data-link-protocol-data-unit.

**DLSAP.** Data-link-service-access-point.

**DLSDU.** Data-link-service-data-unit.

**DP.** Draft Proposal.

**DRDA.** Distributed Relational Database Architecture.

**DTE.** Data terminal equipment.

**FDDI.** Fiber Distributed Data Interface.

**FDOCA.** Formatted Data Object Content Architecture.

**FFT:DCA.** Final-Form Text Document Content Architecture.

**FOCA.** Font Object Content Architecture.

**FTAM.** File Transfer, Access, and Management.

**GDS.** General data stream.

**GOCA.** Graphics Object Content Architecture.

**GOCA EXTENDED.** Extended Graphics Object Content Architecture.

**HDLC.** High-Level Data Link Control.

**IEEE.** Institute of Electrical and Electronics Engineers.

**IIA.** Information Interchange Architecture.

**IOCA.** Image Object Content Architecture.

**IPDS.** Intelligent Printer Data Stream.

**ISDN.** Integrated Services Digital Network.

**ISO.** International Organization for Standardization.

**JTC1.** ISO/IEC Joint Technical Committee 1.

**LAN.** Local area network.

**LAPB.** Link Access Procedures—Balanced.

**LEN.** Low Entry Networking.

**LLC.** Logical Link Control.

**LLC-PDU.** Logical-link-control-protocol-data-unit.

**LLC-SDU.** Logical-link-control-service-data-unit.

**LU.** Logical Unit.

**MAC.** Medium Access Control.

**MAC-PDI** Medium-access-control-protocol-data-unit.

**MAC-SI** Medium-access-control-service-data-unit.

**MACF.** Multiple-association-control-function.

**MAN.** Metropolitan area network.

**MO:DCA.** Mixed Object Document Content Architecture.

**MTA.** Message transfer agent.

**NET.** Network entity title.

**NPDU.** Network-protocol-data-unit.

**NSAP.** Network-service-access-point.

**NSDU.** Network-service-data-unit.

**OMA.** Object Method Architecture.

**OSI.** Open Systems Interconnection.

**OSN.** Office system node.

**PCI.** Protocol control information.

**PDAD.** Proposed Draft Addendum.

**PDU.** Protocol-data-unit.

**PDV.** Presentation-data-value.

**PPDU.** Physical-protocol-data-unit.

**PPDU.** Presentation-protocol-data-unit.

**PSAP.** Physical-service-access-point.

**PSAP.** Presentation-service-access-point.

**PSDU.** Physical-service-data-unit.

**PSDU.** Presentation-service-data-unit.

**PTOCA.** Presentation Text Object Content Architecture.

**RFT:DCA.** Revisable Form Text Document Content Architecture.

**RPC.** Remote procedure call.

**SACF.** Single-association-control-function.

**SAP.** Service-access-point.

**SDLC.** Synchronous Data Link Control.

**SDU.** Service-data-unit.

**SNA.** Systems Network Architecture.

**SNA/DS.** SNA Distribution Services.

**SNA/MS.** SNA Management Services.

**SNAcP.** Subnetwork Access Protocol.

**SNDCP.** Subnetwork Dependent Convergence Protocol.

**SNICP.** Subnetwork Independent Convergence Protocol.

**SPDU.** Session-protocol-data-unit.

**SQL.** Structured Query Language.

**SSAP.** Session-service-access-point.

**SSDU.** Session-service-data-unit.

**SVC.** Switched virtual circuit.

**TCP/IP.** Transmission Control Protocol/Internet Protocol.

**TPDU.** Transport-protocol-data-unit.

**TR.** Technical Report.

**TSAP.** Transport-service-access-point.

**TSDU.** Transport-service-data-unit.

**UA.** User Agent.

**VC.** Virtual call.

**WAN.** Wide area network.

# CONTENTS