

INFORMATION WAREHOUSE IN THE FINANCE INDUSTRY

Document Number GG24-4340-00

August 1994

International Technical Support Organization
San Jose

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page xiii.

First Edition (August 1994)

This edition applies to the following products:

- DataPropagator Relational Version 1 Release 1, Program Number 5622-244
- DataHub/2 Version 1 Release 1, Program Number 5667-134
- DataGuide/2 Version 1 Release 1, Program Numbers 5622-487 and 5622-488
- FlowMark Version 1 Release 2, Program Number 5621-290
- DataPropagator NonRelational Version 1 Release 2, Program Number 5696-705
- DataRefresher Version 3 Release 1, Program Number 5696-703
- Visualizer Query Version 1 Release 0, Program Number 5871-BBB
- S/390 Parallel Query Server

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for readers' feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 471, Building 070B
5600 Cottle Road
San Jose, California 95193-0001

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1994. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Abstract

This publication is one of three publications that relate Information Warehouse architecture and products to industry applications and requirements. These three publications are:

- *Information Warehouse in the Finance Industry*
- *Information Warehouse in the Insurance Industry*
- *Information Warehouse in the Retail Industry.*

The publications describe the *Information Warehouse Architecture I* and emphasize the following products:

- DataPropagator Relational
- DataHub
- DataGuide
- FlowMark
- DataPropagator NonRelational
- DataRefresher
- Visualizer.

These products provide a variety of functions defined in *Information Warehouse Architecture I*.

This publication is intended for business analysts acting as consultants to an Information Warehouse implementation project and technical professionals who are designing Information Warehouse solutions in the Finance industry. A knowledge of the Information Warehouse framework is assumed.

DS

(118 pages)

Contents

PART 1. INTRODUCTION	1
Chapter 1. Industry Library Introduction	3
1.1 Library at a Glance	3
1.2 Terminology	4
1.3 Introduction to Solution Threads	4
PART 2. THE BUSINESS VIEW	9
Chapter 2. Finance Industry Perspective	11
2.1 Finance Industry Trends	12
2.1.1 Business Success Factors	13
2.2 Industry Challenges	15
2.2.1 Deregulation in a Global Economy	15
2.2.2 Competitive Pressures	16
2.2.3 Depressed Economies	17
2.2.4 Summary	17
2.3 Information Technology in the Finance Industry	17
2.3.1 Business Networking	17
2.3.2 The Integration Imperative	19
2.4 Key Systems	20
2.4.1 Customer Information	20
2.4.2 Risk Analysis	21
2.4.3 Profitability Analysis	21
2.4.4 Asset and Liability Information	21
2.5 Information Warehouse Framework	21
Chapter 3. Business Requirements	23
3.1 Finance Industry Example	23
3.2 Key Information Systems and Technology	24
3.2.1 Application Development	25
3.2.2 CASE Technology	26
3.2.3 The Work Group Environment	27
3.2.4 Information Catalog	27
3.2.5 Very Large Databases	27
3.2.6 Query Systems	28
3.2.7 Historical Data	28
3.2.8 Network Transparency	29
3.2.9 Data Replication	30
3.2.10 Requirements Summary	30
PART 3. THE TECHNOLOGY VIEW	31
Chapter 4. Financial Application Architecture	33
4.1 The Architecture Structure View	36
4.1.1 The Application Layer	37

4.1.2 The System Layer	41
4.2 The Enterprise Environment View	41
4.2.1 The Information Model	42
4.2.2 The Development Environment	45
4.2.3 The Network	46
4.3 Financial Services Data Model	46
Chapter 5. Information Warehouse Framework	49
5.1 Value of the Information Warehouse Framework	50
5.2 Why Data Replication	51
5.2.1 Operational Systems	51
5.2.2 Database Technology	52
5.2.3 Cost of Data Access	52
5.2.4 Historical Data	53
5.2.5 Ownership	53
5.2.6 Point-in-Time Data	53
5.2.7 Reconciliation	53
5.3 The Information Warehouse Architecture	53
5.4 Using the Information Warehouse Architecture	55
5.5 Access Enablers	56
5.5.1 Embedded SQL	58
5.5.2 SQL Call Level Interface	58
5.5.3 Distributed Relational Database Architecture	59
5.6 The Finance Industry	59
5.6.1 Information Warehouse Architecture Goals	60
5.6.2 Information Warehouse Architecture Focus Areas	60
5.7 Data Replication	61
5.7.1 Copy Tool Usage	61
5.7.2 Single Point of Control	61
5.7.3 Interface Orientation	62
Chapter 6. Finance Solution Thread Overview	65
6.1 Business Assumptions	66
6.2 The Solution Thread	67
6.3 The Business Function	69
6.4 Information Requirements	70
6.4.1 Customer Information	70
6.4.2 Profitability Analysis	72
6.5 Data Replication Strategy	72
6.5.1 Minimize Systems Administration Workload	73
6.5.2 Leverage Investment in Products and Strategy	73
6.5.3 Minimize Data Copying Cost	73
6.6 System Configuration	74
6.6.1 Platform Configuration	75
6.6.2 Communications Configuration	75
Chapter 7. Organization Asset Data	77
7.1 Foreign Currency and Traveler's Checks Model	79
7.1.1 Logical Data Model	80
7.2 The Entities	81
Chapter 8. Data Replication Tools	83
8.1 Data Replication Type Requirements	85
8.1.1 Business Requirement	85
8.1.2 Update Propagation	87
8.1.3 Copy Consistency	87
8.1.4 Update Sequence	88
8.2 Data Replication Technologies	90
8.2.1 Data Access Protocol	91

8.2.2 Update Propagation	91
8.2.3 Refresh Propagation	91
8.2.4 Archive	91
8.3 Data Replication Products	92
8.3.1 DataHub	92
8.3.2 DataPropagator Relational	92
8.4 Implementing the Solution Thread	103
Chapter 9. Conclusions	105
Appendix A. Models and Modeling	109
A.1 The Construction Model	109
A.1.1 Entity: Things	110
A.1.2 Entity: Agreements	110
A.2 The Annual Report As a Model	111
A.3 Information Warehouse and Modeling	111
List of Abbreviations	113
Index	115

Figures

1.	Basic Set of Business Objects	6
2.	Reach and Range	20
3.	FAA Structure	35
4.	FAA: Application and System Layers	36
5.	FAA Enterprise Application Framework	39
6.	The Data Dimension	41
7.	FAA Information Model	42
8.	The FAA Submodels	44
9.	Information Warehouse Architecture	54
10.	Information Warehouse Framework	56
11.	Access Enablers	57
12.	DataHub Environment	62
13.	The Foreign Currency Transaction	68
14.	Two-tiered Customer Informational Database	69
15.	Hardware Configuration	75
16.	Organization Asset Data	78
17.	Business Data Model	80
18.	Copy Tools	83
19.	Transaction Time Line	88
20.	Stock Pick Transaction Sequence	90
21.	DataPropagator Relational Components	94
22.	DataPropagator Relational Logical Servers	96
23.	Changed Data Capture	99

Tables

1. Library of Information Warehouse: Products Covered	4
2. FAA Submodels Implementation	45
3. Data Placement	82
4. DataPropagator Relational Propagation Paths	93
5. DataPropagator Relational Components and Products	94

Special Notices

This publication is intended to help:

- Business analysts understand Information Warehouse (IW) architecture concepts
- IBM technical professionals understand industry environments
- Customer data processing personnel understand industry environments.

The information in this publication is not intended as the specification of any programming interfaces that are provided by a variety of products that perform functions described in the Information Warehouse architecture. See the PUBLICATIONS section of the IBM Programming Announcement for these products for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

BookManager	CICS/ESA
Common User Access	CUA
DataGuide	DataHub
DB2	DB2/2
DB2/6000	IBM
ImagePlus	IMS/ESA
MVS/ESA	PS/2
QMF	RISC System/6000

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

Apple	Apple Computer, Inc.
BRIDGE/FASTLOAD	Bridge Technology, Inc.
KnowledgeWare Application Development Workbench	KnowledgeWare, Inc.
MacIntosh	Apple Computer Company
Microsoft, Windows	Microsoft Corporation
Motif	Open Software Foundation
OSF	Open Software Foundation, Inc.
ObjectStore Database	Object Design, Inc.
1-2-3, Lotus, Freelance, Freelance Graphics	Lotus Development Corporation.
OMegamon	Candle Corporation
Fast Load	PLATINUM Technology Inc.
Fast Unload	PLATINUM Technology Inc.
Rapid Reorg	PLATINUM Technology Inc.
Quick Copy	PLATINUM Technology Inc.
In2itive	LEGENT

Other trademarks are trademarks of their respective companies.

Preface

This document is intended to merge industry analysis, industry architecture, the Information Warehouse architecture, new product discussion, and specific solutions to industry requirements. It contains discussion of specific industry issues, industry architecture for data processing, Information Warehouse architecture, and solutions.

This document is intended for business analysts and data processing professionals.

How This Document Is Organized

The document is organized as follows:

- Introduction

This part introduces the library within which this particular book is included.

- The Business View

This part establishes the business-oriented level set from which business requirements and Information Warehouse solutions are developed. The first chapter presents a perspective on the Finance industry that includes trends and challenges, key systems, and information technology's position in the Finance industry.

- The Technology View

This part presents the technology solutions for the business requirements established in the business view. It includes an overview of the industry application architecture and the Information Warehouse architecture. It then discusses the individual components of the Information Warehouse architecture and the solutions to those components, according to the needs of the industry.

Related Publications

The following publications are considered particularly suitable for a more detailed discussion of the topics covered in this document:

- “An Architecture for a Business and Information System,” B.A. Devlin and P.T. Murphy, *IBM Systems Journal*, Vol. 27, No. 1 (1988)
- “Building Business and Application Systems with the Retail Application Architecture,” P. Stecher, *IBM Systems Journal*, Vol. 32, No. 2 (1993)
- *Client-Server Computing: The Design and Coding of a Business Application*, GG24-3899-00.
- *DataGuide/2 V1: Using DataGuide/2*, SC26-3365
- *Delivering Data to the Information Warehouse*, Rob Goldring, InfoDB Summer 1992
- *Financial Application Architecture: FAA Concepts of Application and System Architectures*, LY38-4402-0
- *Information Technology and the Management Difference: A Fusion Map*, *IBM Systems Journal*, Vol. 32, No 1 (1993)
- *Financial Application Architecture Introduction*, GC31-3932-0
- “The Future of Health Care Information Systems,” Michael Carrigan, *Hospital Materiel Management Quarterly*, August 1993
- *Information Warehouse Architecture I*, SC26-3244
- *Insurance Industry Futures: Directions for the 21st Century*, Anderson Consulting and LOMA 1993
- “Loaning Banks Some Courage,” *Information Week*, August 12, 1993
- “The Model Business,” *IW Today*, August 12, 1993
- *Principles of Life and Health Insurance*, G. Morton, 1988 LOMA
- “The Spectrum of Data Delivery for Business Information Systems,” Rob Goldring, *DB/Expo92*

International Technical Support Organization Publications

- *Information Warehouse Architecture and Info. Catalog*, GG24-4019
- *Information Warehouse Storage Management Guidelines and Considerations*, GG24-4336
- *Information Warehouse in the Insurance Industry*, GG24-4341
- *Information Warehouse in the Retail Industry*, GG24-4342
- *Library for System Solutions: Data Reference*, GG24-4103

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

Bibliography of International Technical Support Organization Technical Bulletins, GG24-3070.

To get a catalog of ITSO technical publications (known as “redbooks”) online, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

How to Order ITSO Technical Publications

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their local IBM office.

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain books on a variety of products.

Acknowledgments

The advisor for this project was:

Steve Schaffer
International Technical Support Organization, San Jose

The authors of this document were:

Normand Brin
IBM Canada

Wojeich Zagala
IBM Australia

Steve Schaffer
International Technical Support Organization, San Jose

This publication is the result of a residency conducted at the International Technical Support Organization, San Jose.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Paul Englefield, IBM Warwick
Rob Goldring, IBM SWS, Santa Teresa
Eileen Hiltbrand, IBM US
Jacques Labrie, IBM SWS, Santa Teresa
Bill Martin, IBM US
Mark Mauriello, IBM Charlotte Finance Industry
Rita Neuberg, IBM Charlotte
Bill Payne, IBM Charlotte Insurance Industry

Thanks to the following people for reviewing this document:

Thomas Bilfinger, IBM ITSO, San Jose
Don Cameron, IBM ITSO, San Jose
Don Murray, IBM US
Ralph Naegeli, IBM Switzerland
Tom Romeo, IBM US
Michele Schwartz, IBM SWS, Santa Teresa

Special thanks to Ueli Wahli for developing the tool to generate margin comments.

Part 1. Introduction

Chapter 1. Industry Library Introduction

This volume is one of three that look at Information Warehouse architecture and products in the finance, insurance, and retail industries. The three studies yielded somewhat different results but are presented in a standard structure. This introductory chapter describes the library, so that it may be used easily and most effectively by the individual.

1.1 Library at a Glance

The study of the finance, insurance, and retail industries has been made available as a library of three books. Each book takes the same approach to discussing the respective industry, though there is some variation in the aspects of the Information Warehouse architecture covered in each industry. Table 1 on page 4 gives an overview perspective of this library. The library's structure is based on a common set of topics that are addressed consistently across the industry studies, and different aspects of the Information Warehouse architecture being addressed in each study. This structure minimizes redundancy. Therefore, a complete review of Information Warehouse architecture function and products may require reading of all three industry studies.

The common set of topics presents the study flow from a high-level perspective of the industry down to the discussion of the Information Warehouse product technology. These topics are broken down into the business view and technical view as follows:

- Business view
 - Industry perspective
 - Business requirements
- Technology view
 - Industry application architecture
 - Information Warehouse architecture
 - Information Warehouse framework components.

Table 1. Library of Information Warehouse: Products Covered	
Industry	Product
Finance	DataPropagator Relational
Insurance	FlowMark Visualizer
Retail	DataGuide* S/390 Parallel Query Server

1.2 Terminology

The finance, insurance, and retail industry studies address general requirements of the respective industry. They are not studies of a real-world or a contrived enterprise. Rather, they are studies of industry issues and requirements put in the context of an industry enterprise. For this reason, we use the term financial enterprise, insurance enterprise, and retail enterprise, respectively, to reflect this approach.

The knowledge worker is the focus

We begin our study by identifying the knowledge worker as the primary focus of Information Warehouse technology. Knowledge workers are the individuals in an enterprise who make decisions. They exist at every organizational level and have one thing in common: they all need information to make decisions. They get the information through informational applications, also called executive information systems, decision support software, and decision support tools. Informational applications are used to display information provided by the data replication products in the Information Warehouse solution. Therefore, the objective of the studies is to describe knowledge worker's business need for information and the addressing of that need through the Information Warehouse architecture and products.

1.3 Introduction to Solution Threads

Solution thread: applying technology to a problem

A solution thread is a vehicle for applying architectures and products to a generic requirement of the industry. It is a generalized approach, in that the Information Warehouse architecture it uses is generalized for a given data processing function (for example, decision support) across industries, and the industry architecture it uses is generalized for enterprises within a given industry. The products used by the solution thread are generalized for that data processing function, rather than for a business requirement or hardware platform. The solution thread meets the business requirement by customizing the product to the needs of the business.

The Information Warehouse architecture is a generalized architectural approach to managing data and information in a complex business environment. The industry architectures—financial, insurance, and retail—discussed in the three books of this library represent structured approaches to analyzing specific business environments. This analysis leads to specification of requirements, expressed in business terms, which data processing technologies must address. This library presents the Information Warehouse architecture as an architected approach to data processing functions required across the industries, and across the lines of business within each industry. The value of using both the industry-specific architectures and the generalized data processing architecture (Information Warehouse architecture) is to leverage the Information Warehouse architecture functions across the lines of business and to leverage the resources already invested in the industry-specific architectures.

The Information Warehouse architecture is a generalized approach

Not all features of the Information Warehouse architecture nor every product is considered in each of the three industry studies chosen for this library. Review of the three studies as a set will provide information on most of the Information Warehouse architecture features and Information Warehouse framework products. We discuss IBM's published Information Warehouse architecture as a template for connecting business requirements to actual technical implementations, including products plugged into an Information Warehouse framework.

The following example taken from the insurance industry illustrates the leverage of the Insurance Application Architecture (IAA) together with the Information Warehouse architecture. Figure 1 on page 6 shows a set of business objects for the insurance industry. These business objects are used as examples of modeling entities—OBJECT, AGREEMENT, and DEMAND_FOR_DELIVERY—found in the IAA model. IAA defines a modeling entity called an OBJECT. This entity is used to symbolically represent anything that can be insured, such as a life. IAA defines another modeling entity called Agreement. We could use this modeling entity to represent the physical insurance entity called Policy in either personal or property and casualty insurance. This is an example of the generality of the IAA being leveraged across lines of business. We could further use another entity, called DEMAND_FOR_DELIVERY, to represent Claims and PREMIUMS for Premiums.

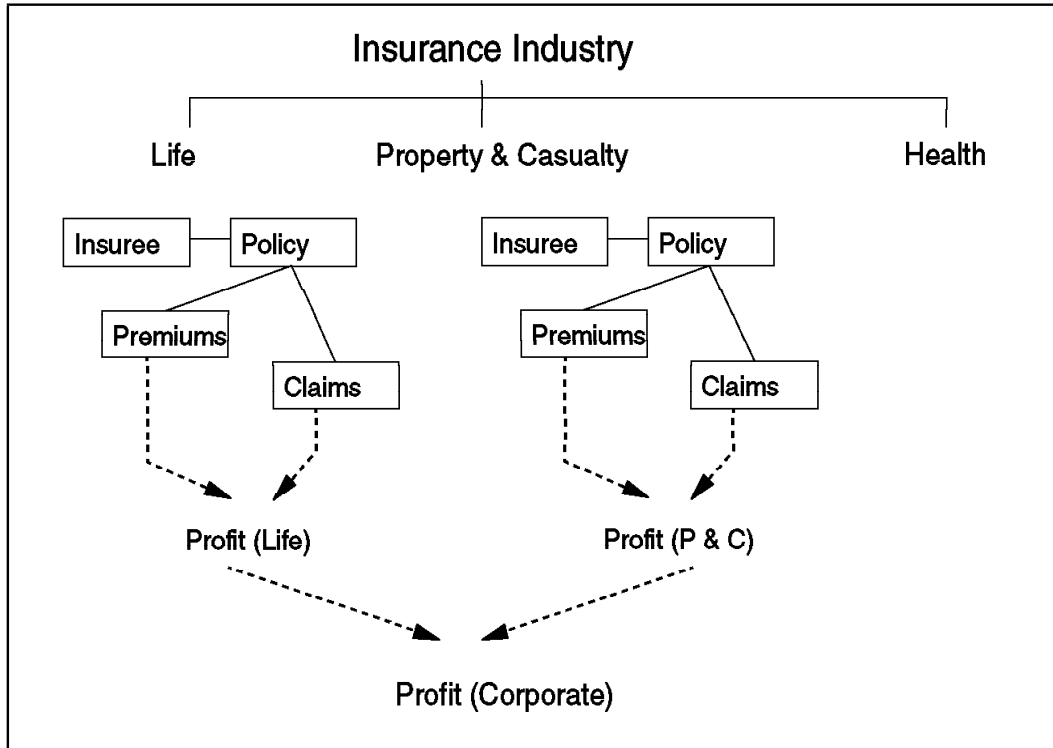


Figure 1. Basic Set of Business Objects. Dashed arrows indicate data flow.

These basic entities correspond to the real-world objects. The Claims entity corresponds to the many claims made by insurees. The Premiums entity corresponds to the many premium payments made by insurees. These claims and payments are implemented as records in an operational database, say, IMS/DB, DB2*, or VSAM. The prime concern of the insurance company is profitability. In this simple exercise, profitability is defined as total premiums minus total claims.

Assuming claims records and premiums records are kept in separate databases, there is a need to bring those two sets of data together. There is an additional need to reconcile this data as it is brought together. The Information Warehouse architecture defines the process by which this can be done in an architected manner. This architected approach to extracting data is called data replication. Data replication is generalized, so that it can be a solution for bringing together Claims and Premiums data for Life insurance, Property and Casualty, or any other insurance product that takes in premiums and pays out claims.

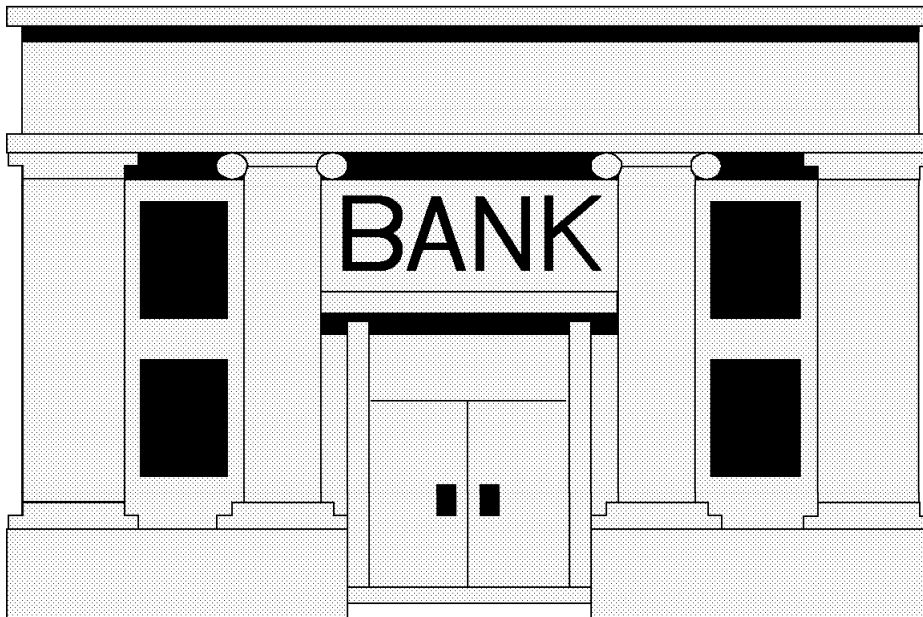
The Information Warehouse architecture provides guidance for data access as well as data replication functions. Data access applies to operational data or informational data copies of the operational data, generated through data replication. Access to operational data (*direct access*) or informational data shares certain requirements for implementation. These requirements can be best understood in terms of the business requirements for data access. The lines of business are assumed to have their own data stores containing records of insurees, sometimes called client files. The Life insurance line of business has an interest in using the client file owned by the property and casualty line of business for prospecting purposes, and vice versa. This requirement brings up two issues relevant to the Information Warehouse

framework: what data is available, and how to access it. The first requirement is resolved through the Information Catalog, while the second is resolved through access enablers or data replication. The point here is that both lines of business have the same needs to access data, and both needs can be resolved by solutions based on the Information Warehouse architecture.

Part 2. The Business View

Chapter 2. Finance Industry Perspective	11
2.1 Finance Industry Trends	12
2.1.1 Business Success Factors	13
2.2 Industry Challenges	15
2.2.1 Deregulation in a Global Economy	15
2.2.2 Competitive Pressures	16
2.2.3 Depressed Economies	17
2.2.4 Summary	17
2.3 Information Technology in the Finance Industry	17
2.3.1 Business Networking	17
2.3.2 The Integration Imperative	19
2.4 Key Systems	20
2.4.1 Customer Information	20
2.4.2 Risk Analysis	21
2.4.3 Profitability Analysis	21
2.4.4 Asset and Liability Information	21
2.5 Information Warehouse Framework	21
Chapter 3. Business Requirements	23
3.1 Finance Industry Example	23
3.2 Key Information Systems and Technology	24
3.2.1 Application Development	25
3.2.2 CASE Technology	26
3.2.3 The Work Group Environment	27
3.2.4 Information Catalog	27
3.2.5 Very Large Databases	27
3.2.6 Query Systems	28
3.2.7 Historical Data	28
3.2.8 Network Transparency	29
3.2.9 Data Replication	30
3.2.10 Requirements Summary	30

Chapter 2. Finance Industry Perspective



In this study, we focus on the finance industry and the roles that the Financial Application Architecture* (FAA), the Information Warehouse architecture, and Information Warehouse products play in that industry. In particular, we present the benefit of addressing the informational needs of the industry in a generalized, architected way. The discussion of the finance industry and its needs for the Information Warehouse architecture starts with a perspective on the finance business. This initial look at the business issues of the finance industry is presented as an overview of the trends and key systems of the industry. In subsequent chapters, we present the business requirements of the finance industry in regard to information systems and then map these requirements to information systems technology. We then present a solution to these requirements based on Information Warehouse architecture and products.

The IW solution draws from industry challenges

This study focuses on a bank as an example of a finance industry enterprise. We assume that all banks, investment houses, securities firms, and even insurance enterprises deal with financial instruments of some sort. Our focus on a bank does not imply that the general solutions suggested are relevant to banks only; the solutions are relevant to any finance enterprise. The solution thread is a generic scenario, representing a conceptually relevant finance industry situation. The Information Warehouse solution incorporates

the Information Warehouse architecture and products as a way of responding to the concerns of the solution thread.

Information technology separates leaders from followers

Information technology is a key factor setting apart industry leaders from others in the field. Industry leaders have gained competitive, organizational, and economic advantage from their information technology investment. On the other hand, some organizations question the return on the investment in information systems.

The rules of the game are changing

The resolution to this dilemma—why some have succeeded and some have failed—has brought the relationship between the business and information technology organizations under careful scrutiny. That scrutiny is carried out in the context of the general atmosphere of the industry. What emerges is the realization that conditions external to the financial enterprise have changed dramatically; markets, competition, and the legal framework are all being redefined. This is happening at a time when advances in technology changed the ground rules in terms of expanding business and enterprise cost structure.

Networking, data, and information: key players in information technology

We discuss the challenges of the industry as a prelude to a discussion of business networking's role in information acquisition. As part of this discussion, we distinguish between data and information and their roles in information systems. Data is traditionally associated with transactions systems, whose mission is to support the day-to-day operations of the enterprise. Information is generally a reconciled and enhanced copy of the data, designed for support of strategic business decisions. The creation of information from distributed heterogeneous data stores requires skilled staff working in concert with various line-of-business personnel.

The information systems department will provide information

The information systems department's mission will evolve from being a provider of operational systems to being a provider of architected informational data. We present guiding principles toward achieving that new mission. We stress the importance of integration and the principles of the Information Warehouse architecture. Industry analysis leads to identification of critical business success factors which, in turn, define key information systems.

2.1 Finance Industry Trends

The focus moves toward the customer

Industry leaders respond to the changing market by providing customer-oriented, market-driven, low-cost products and services. This strategy calls for a close relationship with the customer. Financial institutions need to lock their customers into as many services as possible, both to ensure customer loyalty and to gain additional revenue from their customers.

The finance industry therefore needs solutions that:

- Increase profits
- Improve customer service
- Improve efficiency and productivity of personnel
- Improve ability to sell additional products to existing customers
- Reduce costs
- Migrate customers to self-service
- Improve the quality of loans

- Improve staff training
- Minimize errors and fraud.

In line with these demands, branch organizations have a vital role to play as branch personnel have direct customer contact. Financial institutions are expecting a more marketing approach from their staff and are beginning to equip them with the tools to support this new approach.

2.1.1 Business Success Factors

Finance enterprises have focused on the following critical business factors as key to their success in the future:

- Balance sheet management

Nonperforming assets, particularly real estate loans, present a problem for many financial institutions. Strategies to improve balance sheet position include:

- Improved loan origination process

Few financial institutions have reengineered their lending origination applications. Better access to customer information can improve the quality of credit decisions.

- Increased loan securitization and syndication

Holding assets on balance sheets through maturity is now seen as a potential exposure as long-term loan performance has become difficult to predict. Many banks remove loans from their balance sheets through securitization or reduce exposure through loan participations or syndications.

- Improved credit risk management

Ability to monitor quality of a balance sheet is needed. Risk management systems can identify large concentrations of potential problem loans and enable management to take action before serious problems can develop.

- Product diversification

There is a move among financial institutions to decrease their dependency on balance sheets for revenues and profits by emphasizing fee-based services. Some institutions would immediately sell the loans to remove risk from their balance sheets but use their relationship with the loan customers to sell additional fee-based services.

Customers, on the other hand, increasingly demand more function and integration between products and services. Customers want demand deposit accounts, payment services, lines of credit, and securities accounts linked together. The market demand creates opportunity for differentiation between the institutions.

- Customer relationship management

Existing deposit and loan customers are primary candidates for new fee-based services. Relationship management requires strategies for improved customer service and marketing at a reasonable cost.

- Customer service

Most customer interactions with the financial institutions are service interactions. Retail customers, for example, interact with their banks through ATMs and their monthly statements, which are both service mechanisms. Customer satisfaction would then depend on accessibility, availability, simplicity, responsiveness, and accuracy.
- Sales and marketing

Customer contact personnel are essential to identify and capitalize on marketing opportunities. To do this effectively staff needs information systems. In particular, having access to a history of the customer relationship assists in targeting new products and services.
- Market segmentation

Cost-effective marketing is based on understanding the customer population, which applies to both current and prospective customers. Knowing market segments allows for product development, pricing, marketing, and support—all targeted to a particular market segment.
- Alternative delivery channels

Electronic service delivery (for example, home banking) is one example of alternative methods of reaching customers; it reduces cost and gives the finance enterprise better market penetration.
- Measurements

Effectiveness measurement creates a feedback mechanism that helps in adjusting methods of operation and understanding profit and revenue in different market segments.
- Cost control

Competitive pressures and expected sluggish growth in the economy necessitate close attention to costs. Dominant strategies in cost reduction are:

 - Back office consolidations

The finance industry went through a wave of mergers, often justified by the improved efficiencies of the larger combined organizations. Now institutions are reducing the number of operation centers, branches, and employees.

A related development examines whether a critical mass in any product or market has been achieved. If critical mass (for example, in credit card or mortgage assets) cannot be achieved, it is outsourced, or assets are sold. Having critical mass allows cost effective application of technology.
 - Product rationalization

Converging similar offerings reduce both marketing and operational expenses.
 - Paper elimination

Handling paper is becoming expensive when compared with electronic processing of documents (and optical storage). Many insti-

tutions are striving for a paperless office, or at least handling paper only once, at a point of capture. Following the success of credit card networks, many finance enterprises are investing in image technology.

– Branch office operations

Financial institutions are a leading employer, with the majority of their staffing being branch workers. Branch organization structure is one of the largest expense centers in banks. The effectiveness of branches, with the high cost of staff, real estate, and other facilities, is questionable, when measured as either a marketing and service delivery channel or supplier of funds. Cost reduction is achieved by consolidations and further investment in branch automation.

2.2 Industry Challenges

Industry research has shown that financial institutions around the world are experiencing similar difficulties, including a multitude of environmental challenges. We have grouped these challenges into general categories, as follows:

Finance enterprises face similar difficulties

- Deregulation in a global economy
- Competitive pressures
- Depressed economies.

2.2.1 Deregulation in a Global Economy

Government and financial institutions are no longer the autocratic drivers of the finance industry; control of the finance industry has shifted to the customer. Financial institutions are becoming more and more reactors to the environment than controllers of it. This change is caused by smarter, more demanding customers and a freer economic climate.

Customers drive the business

Today's financial customers are smarter today than ever before. They have greater access to basic financial information as well as sophisticated analyses of that basic financial information. They can easily bypass financial institutions in seeking access to capital and price information. Thus, customers demand more in their financial services. They demand products more tailored to their needs, direct access to funds and account information, and higher quality service.

Customers are smarter and more demanding

Deregulation has had the biggest impact on financial institutions and will continue to do so throughout the 1990s. Deregulation began in the early 1980s and has resulted in the loosening of geographic boundaries on the flow of capital and labor. Financial institutions were also allowed to enter business areas from which they were previously barred. For example, commercial bank holding companies in the United States can now provide mortgage banking services.

Deregulation regulates the industry

*Successful
finance enter-
prises
assume a
global
economy*

Deregulation, mergers, acquisitions, and joint ventures have become the norm in the financial services industry and are global in scope. Financial institutions must now deal with instability created by varying inflation and exchange rates, changing government, regulations, and global political events.

Deregulation and subsequent mergers and acquisitions have put pressure on financial enterprises to consolidate, integrate, and manage information systems organization, hardware, and software resources. The ability to do so is a key success factor for the newly created consolidation. The integration is likely to be a costly and difficult task, given the complexity and heterogeneity of the original systems.

2.2.2 Competitive Pressures

The environment within which finance enterprises operate has become much more competitive. The competition is coming from both traditional and non-traditional sources. These pressures appear in the number of players in the industry and in the products they are competing to market.

*Deregulation
and consol-
idation
increase com-
petition*

Deregulation and consolidation within the industry has raised the level of competition; nonbanking institutions have entered the financial services marketplace. For example, General Motors** and AT&T** corporations have entered the lucrative field of extending credit by offering their own credit cards, tied loosely to their primary, core businesses of selling cars and communications service.

*All products
are commod-
ities*

Diversification and differentiation of products emerge as key responses by leading financial institutions to the loss of exclusive markets and the transition of financial products into a commodity market.

*Information
technology is
a competitive
tool*

Competitors and customers expand use of new technologies in their organizations and create pressure on financial institutions to do the same and respond with competitive services, products, and reduced costs. The finance industry is heavily oriented toward information. Financial instruments become more information-based and less concrete as they become more sophisticated. Some financial instruments exist only as they are represented by information and have little or no direct representation as a traditional financial object. Common stock is an example of a traditional financial instrument; index futures are an example of a financial instrument existing only as an information object.

2.2.3 Depressed Economies

There has been a significant increase in nonperforming assets, particularly in the area of real estate loans. Margins on core services are being squeezed because of competition. These factors force finance enterprises to examine their products for the profit they represent. The successful finance enterprise is adept at analyzing profit by product and at keeping successful products while walking away from unsuccessful ones. Sluggish economies will continue to affect loan quality and growth. There is also scarcity of capital, contributing to lower profits and decreasing stock prices of finance enterprises.

A tight economy puts a higher focus on profit margin

2.2.4 Summary

The issues discussed in this section represent business challenges to every enterprise in the finance industry. Some of these challenges can be met directly through information technology, some cannot. Some can be met indirectly with information technology by utilizing informational data made available by the Information Warehouse architecture and products. The successful financial enterprise is one that recognizes the business challenges, understands the potential of information technology, and can apply that potential to the challenges.

Recognize the challenges and apply the technology

2.3 Information Technology in the Finance Industry

The relationship between the business and information technology organizations has changed as a result of the pressures and challenges in the finance industry. Business executives have long sought ways to integrate business and data processing thinking, experience, and strategy. Historically, the information systems strategy was formed as an understanding between the line-of-business and the information systems executives. The strategy was then communicated down the organization structure from the information systems executive to those charged with its implementation. Recent developments suggest a far more interdependent relationship between business planning and information technology at all levels, from each executive down to every level of the respective organizations. This section discusses different aspects of the relationship from a business perspective.

Organizational cooperation at all levels is crucial

2.3.1 Business Networking

Business networking, a phenomenon of the late 1980s, has had a tremendous impact on the finance industry. It combined the massive deployment of computers, information stores, and telecommunication networks to transform the basic workings of the industry. In banking, large networks of automated teller machines (ATMs) emerged to becoming the new core of banking. Cash management and foreign exchange trading systems were similarly reshaped. The transformational experience of networking was not limited to the finance industry.

Business networking transforms business

For the retail industry, it was POS

The retail industry's equivalent was in the emergence of point-of-sale (POS) systems. This led to electronic streamlining of merchandising, ordering, distribution, and inventory control on the operational side of the business. The strategic side saw faster management analysis of operational data, which enabled quicker response to trends and problems. In the airline industry airline reservation systems became the base for marketing, pricing, scheduling, and many aspects of strategic planning.

Business networking means business information

In each instance, business networking created or reshaped a core logistic, that is, a business process fundamental to the basic operations in an industry. Business networking as a core logistic constitutes a fundamental piece of organization and competition within the finance industry. Its data processing parallel is the information systems infrastructure.

Informational analysis is the ultimate goal

The information systems infrastructure—the databases, computers, LANs expert systems software, and data are the *visible* components that support business networking. The ultimate objective, however, is information acquisition. Effective information acquisition requires differentiating between the active process of being informed and the passive process of receiving data. passive data reception is the predictable, predefined collection of data from operational systems; its benefits are limited to the original intent and design of the operational system.

Knowledge workers drive information acquisition

End users working with information (also called knowledge workers) are a key element of being actively informed rather than passively receiving data. The effective knowledge worker initiates the information gathering process, understands the meaning of the information, and uses it for the benefit of the enterprise. The enterprise supports this knowledge worker by presenting operational data in an informational environment so that it can manage and react to transformations in its core logistics. The informational environment implies reconciliation and enhancement of the operational data, presentation of information in appropriate graphic form, and availability of meta-data to describe the informational data.

Information is used to competitively manage the business

Knowledge workers are responsible for actively seeking and analyzing information that was delivered by information systems from operational data sources. Failure to use information competitively and properly is a prescription for disaster. A 50% attrition rate in industry players over the last 10 years is to a large degree attributed to late or no recognition that business networking changed the rules of competition.

Business networking changed the rules of competition.

One example of business networking changing the rules of competition is the emergence of airline reservation systems as a force in the hotel business. Initially, hotels did not consider airline reservation systems to be a factor in hotel reservations. However, the airlines' ability to offer hotel bookings along with airline reservations forced hotels to develop their own reservation systems and form partnerships with airlines as a part of their business strategy. The airlines gained competitive advantage by making the first customer contact and using that position to influence the customer's choice of lodging.

2.3.2 The Integration Imperative

A decentralized business philosophy favors decentralized information systems planning and implementation. This translates into the decentralized technology of programmable workstations, LANs, and departmental systems used to support decentralized operations. This view tends to equate information technology as essentially equivalent to computers. This view does, however, overlook the business trends generated, stimulated, or supported by business networking. Decentralization overlooks the importance of sharing data across products, services, locations, companies, and country organizations. There is also a shift from largely independent business function to interdependent functions, and from product- to relationship-based services and cross-selling.

Decentralization may mask trends

The information technology platform is a common business resource and a base for services delivery. Its business function can be defined in terms of two dimensions: reach and range (see *Information Technology and the Management Difference: A Fusion Map*). Reach describes the locations to which the platform can link. Range describes the information that can be shared directly and automatically across services and systems. Figure 2 on page 20 shows reach and range and their relationship to the business. It shows the various components that make up the information technology resource.

Information technology is a common business asset

The integration of these information technology components is increasingly regarded by industry experts as a cornerstone of business integration. Business integration is the linking of previously independent services and operations for the overall benefit of the enterprise.

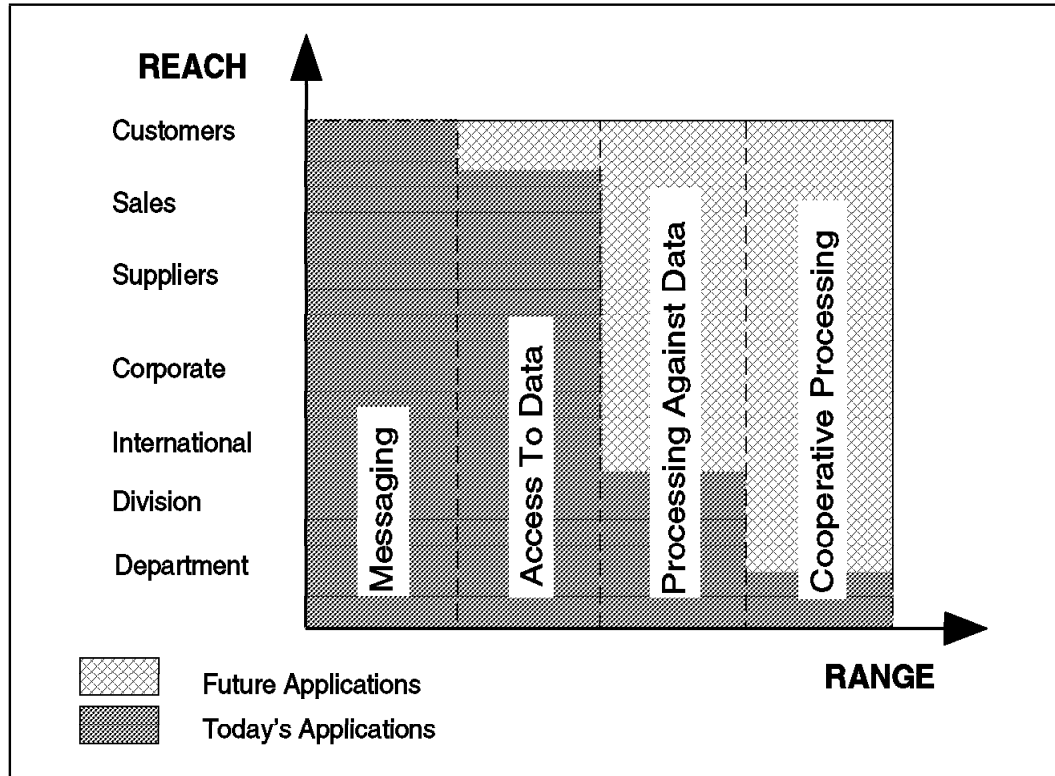


Figure 2. Reach and Range

2.4 Key Systems

Industry analysts cite certain information systems as being key to the enterprise's long-term success. These systems are the following:

- Customer information
- Risk analysis support
- Profitability analysis
- Asset and liability information.

These systems are discussed in the sections that follow.

2.4.1 Customer Information

Customer database is the first step to the client

Access to complete customer information is key to analyzing the performance of different demographic segments of the institution's customer set. This information helps business analysts project the success of new products and services. It also helps shape marketing strategies appropriate to new products and targeted market segments. A customer information database is used to link information residing in different operational systems. This enables a finance enterprise to offer personalized service based on a history of business transacted with a particular customer. This may be a discounted fee for service or a springboard to offer new product, with a high likelihood of acceptance.

2.4.2 Risk Analysis

Financial institutions use risk analysis systems to evaluate exposures of various kinds. Credit risk systems allow institutions to evaluate their lending exposures with respect to economic or geographic sector, customer or loan type, or other characteristics. Operational risk systems allow institutions to evaluate their settlement exposure on their portfolio of transaction services by a variety of categories. The Credit Risk Management System, an American Management Systems product, is an example of such a system. It allows finance enterprises to analyze their credit exposure by parameters such as industry concentration, geographic distribution, and loan type. Risk analysis helps the finance enterprise commit to loans which are most likely to be profitable and avoid those that are likely to be nonprofitable.

Risk analysis minimizes the down side

2.4.3 Profitability Analysis

Profitability analysis systems allow institutions to measure the costs and returns associated with various operations of their organizations. Profitability analysis can vary greatly in detail and timing (for example, once a year versus continuous monitoring or line-of-business versus individual products). We generally assume semicontinuous monitoring on a product level. Profitability of various marketing and sales departments or organizations can also be evaluated. The Earning Analysis System—a joint effort of Pittsburgh National Corporation, Hogan Systems, and IBM—supports profitability analysis by a number of categories including customer, product, and marketing organization.

Profitability analysis identifies better revenue opportunities

2.4.4 Asset and Liability Information

Access to complete information on interest rates and maturity dates of assets can assist institutions to manage the risks associated with changes in interest rates.

2.5 Information Warehouse Framework

The key systems have a common need for functions defined and supported by the Information Warehouse framework. These systems can all be described as informational applications that contribute to the strategic success of the finance enterprise. They all depend on data generated by existing operational applications that contribute to the tactical, day-to-day success of the finance enterprise. The Information Warehouse framework provides the architecture and the products to deliver the operational data to the key systems in an informational form. The key systems then contribute to the better use and direction of the original operational applications. The operational data is delivered through an architected process of extraction, reconciliation, and loading from heterogeneous data stores.

Operational applications feed informational applications, which direct operational applications

Chapter 3. Business Requirements

We use business requirements to describe the needs for data processing solutions to the challenges in the finance industry. These challenges derive from the trends, directions, and pressures of the industry. We assume that the pressures persist over a relatively long period of time and that they are general in nature, rather than specific to an immediate narrow-focused need. Independently developed applications or application systems have been the usual response to specific product, line-of-business, or project level application requirements.

An architecture helps solve strategic problems

We take the view that the solution to a strategic pressure must be done in the context of an architected approach. This approach assumes that multiple applications will be needed over the course of the existence of the business pressure. The architecture presents a standard approach for all applications developed to meet different aspects of the business pressures. The architectures we use are the FAA, to accommodate specifics of the finance industry, and Information Warehouse architecture, to accommodate the cross-industry need to manage informational data and applications. These architectures are used as a guide for developing or acquiring software solutions identified by the architectures.

Business pressures are strategic, as are architectures

3.1 Finance Industry Example

It takes accurate information and knowledge of how to apply that information to achieve higher performance. Sheshunoff** financial information products offer not only the data you need, but the guidance to use this information to your best advantage. Sheshunoff financial information is based on the latest data released by the federal regulators of financial institutions. Every quarter, Sheshunoff analysts run hundreds of edit checks to ensure the accuracy of the data, and present it in a variety of easy-to-use, comparable formats. Sheshunoff Information Systems, Inc. is a service bureau organization that provides information as a revenue-producing product to enterprises in the finance industry.

IW concepts are used to produce real revenue

This finance industry example shows the use of Information Warehouse concepts in a revenue-producing product. The process of acquiring the raw data from public domain, federal organizations is an example of data replication. The source happens to be external to the enterprise and happens to be physically acquired through a phone line or physical tapes delivered, rather than accessed on a direct access storage device (DASD) within the enterprise. The process of edit checking fits into the Information Warehouse architecture description of reconciliation, and the massaging of the data to show different analyses of data is an example of data aggregation and derivation, the result of which is the Information Warehouse architecture's derived data. Finally, the reports as presented in easy-to-use formats exemplify the use of informational analysis to produce a report that is easily understood by the business analyst. It cannot be overemphasized that this is a product that brings in hard revenue for the enterprise.

3.2 Key Information Systems and Technology

Key information systems challenge the technology - business requirements magnify technical challenges

We have identified the key information systems in the finance industry for the 1990s.

These key information systems present a challenge to existing technology. What makes them different—in particular, customer information and profitability analysis—is their scope, complexity, amount of data they need to process, and the cost of development. These issues are discussed as a lead-in to their technology solutions.

Informational systems have a broad organizational scope. This broadness underscores the need for data reconciliation at all levels of the organization. Reconciliation is necessary to make operational data understandable to the business analyst. It is also necessary to reconcile different representations of data from one line of business to another.

Systems need to reference a variety of hardware and software platforms in a heterogeneous environment. They also cross line-of-business boundaries and levels of organization to present the end-to-end picture of the underlying business processes. Some systems such as risk analysis also introduce complex logic to the use of data for informational purposes.

Systems at the detailed, transaction level require large volumes of data for any enterprise that wants maximum return on its information technology investment. Maximizing the return in the largest of enterprises typically drives the need for hardware and software solutions specialized for that need. The data volume issue is exacerbated by the enterprise pursuing historical—trend—analysis.

Systems deployed down to the branch level need a strategic horizon of deployment. The finance industry generally looks at hardware technology having a 10-year life and accepts the need to replace hardware on that schedule. The current example of that cycle is the trend toward the LAN platform in the branches. New models of computing are explored (for example, client-server computing) and are deployed based on their value versus cost. The information technology organization is a key component in this ongoing technology race.

We now look at the customer information, profitability analysis, risk analysis, and asset and liability information systems in terms of the technology they require. The technology required falls into the following software and hardware areas:

- Application development
- CASE technology
- Work group environment
- Information Catalog
- Very large databases
- Query systems
- Historical data
- Network transparency
- Data replication.

The requirements serve as a blueprint for architecting a solution to specific business problems.

3.2.1 Application Development

Application development is an evolutionary discipline. It began with first generation languages and file systems and is moving through an age of relational technology, into the future of object-oriented programming and databases. The general objective is to improve the application development process and make design, generation, testing, and implementation of applications easier and quicker. The evolutionary trend, while simplifying the task of implementing new applications, has created a portfolio of mixed-technology applications. Reengineering of existing applications is, in general, not a good business and economic decision. The enterprise tends to keep existing applications, performing maintenance only when necessary. The information systems department is left with the task of managing a complex combination of programming and data storage technology.

Application development evolution created a complex environment

The continued existence of earlier-technology systems, called legacy systems, poses a more specific problem to the information systems department. These legacy systems are an investment for the enterprise and are often a mission-critical component of the information technology infrastructure. They are difficult and costly to rebuild and may be perfectly acceptable in their current form from functional and performance points of view. However, the data created and maintained by these systems is the source for the enterprise's key informational systems. Current-technology informational applications need to get at the data being held in older technology data formats. The information systems department must find ways to interface the new informational applications to the existing legacy data.

Legacy systems are part of the strategic picture

The evolution continues. The enterprise and its information systems department must continually embrace new technology to stay competitive and survive. It must do so in a cost-effective manner, and in as nondisruptive a manner as possible. These challenges and environmental conditions make it difficult for the finance enterprise to develop the key systems it needs to compete. These technology issues are only one factor; the organizational and business environment factors further complicate the process of developing key systems.

New technology must be integrated

Information Warehouse accommodates application development history

The complexity of application development is accommodated by the Information Warehouse architecture and products. A variety of application development technology exists and will exist in the implementation of operational systems. Information Warehouse architecture and products manage data replication to take operational data from all technology sources in a structured approach. Legacy systems are supported by data replication tools for extraction. The variety of hardware and software platforms is supported by the open interfaces of the Information Warehouse architecture.

3.2.2 CASE Technology

CASE helps manage complexity

Information modeling for application development has two major considerations: the intellectual effort of representation and the integration of that representation with an application generator (a tool that will generate application code). The objective of computer-aided software engineering (CASE) technology is to manage the complexity of this representation and integration environment. The complexity begins with the modeling design effort. Modeling design is often represented at different levels of abstraction within the organization. Data represented once in a conceptual way is likely to exist in two or more different data stores at one time or at different times in its life. An example is a logical database design which is transformed into a physical database design such as relational tables in DB2. Physical database definitions become “hard-coded” in application programs, causing concerns in strategic model and application maintenance. Understanding the impact of change at any abstraction level on the applications and the model itself is a challenge. CASE tools can greatly assist in this task; they open doors to integration.

CASE has its limitations

CASE tools today operate on a technology level, rather than higher levels of abstraction. They capture some limited amount of data semantics and business process logic. They can even generate simple code out of these specifications. However, they are generally not able to tackle complex, context dependent forms of information. The recognized limits on the modeling process and the model to application generator integration process are well known. There does not, however, appear to be alternatives to high-level abstraction—modeling—for managing integration.

We DO need to understand our data

There is a compelling case for business to understand its data and processes in an operational, day-by-day sense. This understanding is a prerequisite for understanding the data and processes in an informational, strategic sense. This can be achieved if information structure is brought together. Until the model to the application generator integration problem is addressed, modeling will continue to serve as a communication vehicle between the information systems department and business analysts.

3.2.3 The Work Group Environment

Business expects information systems solutions to be delivered in a work group environment for maximum flexibility and control by the work group; the LAN is particularly well suited to meet this requirement. LAN-based, local database servers leverage the low cost of hardware with the ability to deploy a multitude of analysis tools. The LAN servers can service a major portion of work group requirements for data and data access response time.

The LAN is the work group platform

The work group mindset loses the perspective of data across work groups, lines of business, or other organizational division. In addition, the increased cost of system support duplicated across work groups is often overlooked. Informational systems that provide access to enterprisewide data must avoid the narrow vision and duplicative costs of the work group.

The work group perspective is limited

The Information Warehouse framework enables knowledge workers to access data from sources throughout the enterprise using a GUI. Traditional large servers used in client-server Information Warehouse implementations lead LAN servers in their ability to process complex queries involving large amounts of data. The large amounts of data are an expected component of analysis of data across work groups and lines of business.

The Information Warehouse framework leverages data and system resource

3.2.4 Information Catalog

Business analysts and knowledge workers spend far too much time finding business data and too little time using it. Having found it, they still spend unproductive time trying to understand the business data. The Information Warehouse framework recognizes the need to manage the process of finding and accessing data. It qualifies the role of meta-data—data about data—in this process, and defines two interfaces that play a role in solving the problem. The Information Catalog in *Information Warehouse Architecture I* discusses the nature of the problem. The Information Catalog API and the Information Catalog import/export interface are the published interfaces for software solutions to this problem. For more complete discussion of these interfaces and the IBM solution based on these interfaces, see *Information Warehouse in the Retail Industry*.

Knowledge workers must first find informational data

3.2.5 Very Large Databases

Financial enterprises tend to have a customer base numbering in the millions. This base number is the multiplier by which the data kept for each individual customer is factored. The different types of data kept for each customer (for example, address and account information) lead to large data volumes. Industry experience suggests that for every million customers, banks would like to maintain 4-5GB of numeric and textual information in their customer information system. This information is currently limited in its historical span to perhaps one previous month's details or year's totals. Adding a full year of detail would probably push the ratio to about 20GB of data per million customers. The minimum requirement for the database is to support textual and numerical information for every customer's account online. Today, this information amounts to gigabytes of storage.

Gigabyte volumes are common

3.2.6 Query Systems

Data access must be flexible

Knowledge workers need flexibility in the analysis queries they formulate and the information against which the queries are executed. The analysis may take two forms: it may be using information to support hypotheses on business activities or it may be using information to discover those truths. The information may be reconciled only and correspond on a row for row basis to the operational data or it may be derived or summarized to a single value, the enterprise total. Knowledge workers may want to use information at any level of summarization between these two extremes, and they may want vastly different levels of summarization from day to day. Knowledge workers using information in these ways are excellent candidates for query systems.

A query system must process gigabyte volumes quickly

The intent of query systems is to address heterogeneity and response time requirements. Heterogeneity implies accessing data on different, unlike platforms, correlating it, and presenting it to the knowledge worker in a common form. Technology barriers rule out the simplistic approach of actually accessing data where it exists. Rather, the likely scenario is copying that data, with reconciliation and aggregation, to a single query system platform. This is a key role of data replication (for more information on data replication, see Chapter 8, "Data Replication Tools" on page 83).

Query systems must process and aggregate huge amounts of data in a reasonable time. Meeting this requirement is a major challenge to Information Warehouse implementation. Current solutions use specialized hardware and software (S/390 Parallel Query Server) or specialized function within generalized solutions (I/O parallelism in DB2 Version 3.1). In either case, parallel processing technology is the key technology in meeting response time demands in a large data volume environment. Low-cost computing technology goes hand-in-hand with a massively parallel computing environment. Other requirements include distributed query capability, defined as the use of a single data access language command request to access multiple data sources. This technology is defined in Distributed Relational Database Architecture (DRDA) as the distributed unit of work.

3.2.7 Historical Data

Historical data is essential for trend analysis

The requirement to maintain historical data pushes data volumes past the 100GB point into the terabyte (10¹² bytes) range for a single database. Existing database and hardware technology is hard pressed to support query against databases of this size. Specific issues associated with accessing databases of this size include the cost and response time in storing and accessing this volume of data.

Storing large volumes of historical data is a major financial expense. The proper approach to this problem takes into account the variety of storage media available. High-speed magnetic disk supports the fastest response time but is most limited in storage capacity and most expensive. Optical disk has elongated response time but is virtually unlimited in storage volume and is relatively inexpensive. Other types of media and different versions of each media type create a continuum of options for managing small and large data volumes. A successful Information Warehouse implementation leverages each type of media according to its cost, capacity, and appropriateness for the use of the informational data. For more information on the use of storage media in an Information Warehouse implementation, see *Information Warehouse Storage Management Guidelines and Considerations*.

Map storage media types to the use of informational data

Response time for accessing historical data is a critical issue. The total time required to perform informational analysis of historical data depends on a variety of factors, including the following:

Response time is critical for trend analysis

- Processor speed
- Storage device access time
- Volume of data being accessed
- Query complexity.

Processor speed and storage device access time are fairly predictable and constant. Data volume and query complexity, by definition of decision support function, are constantly changing. These factors combine to make response time prediction for informational analysis of historical data very difficult. Historical data analysis must be viewed on a sliding scale with respect to the total response time. Acceptable response time for this activity may range from seconds for simple queries on active data to minutes on recently aged data to hours for query on very old data.

Financial institutions are moving to image technology. The volume requirement for historical data is as much a factor for image data as for other types of data. A page of high resolution, print quality graphics may take as much as 400MB of storage (as opposed to 8K-30K for text). Image data, however, is not as volatile as text or numeric data. The image data concerns put more focus on the need to map the use of data to the storage media.

Image data is a newly important form of information

3.2.8 Network Transparency

The communication network is also a major investment. Finance enterprises can ill-afford the cost of redundancy in their communication infrastructure. Yet, this redundancy tends to be the rule rather than the exception. There is no common communication protocol, nor is one expected to emerge in the near future. Network protocol toleration is necessary to manage the different protocols in an enterprise.

Network cost is significant

DRDA eases network complexity

IBM developed DRDA to manage connectivity issues associated with distributed relational databases. DRDA is a highly efficient, open protocol that takes advantage of other published architectures, including Systems Network Architecture. It has grown to include TCP/IP in the UNIX environment. This connectivity extends to the UNIX (AIX) environment through DB2/6000's use of the protocol to access data on other platforms such as DB2 for MVS. Tolerant of other protocols (for example, TCP/IP by SNA) is a step toward meeting the network transparency objective.

3.2.9 Data Replication

Data Replication is essential to informational needs

The informational needs of the finance industry demand a leveraged use of data replication. Data replication is a fundamental issue in this study and is covered in depth in Chapter 8, "Data Replication Tools" on page 83.

3.2.10 Requirements Summary

Information Warehouse architecture is a generalized way of dealing with requirements

Developing the key information systems for the finance industry is a formidable challenge for the information systems department. Some of the challenges demand new technologies. Overall, a generalized, architected approach is needed to facilitate integration and reduce cost and complexity.

Information Warehouse Architecture I defines a reference structure and a set of formats, protocols, and interfaces that can be used to build an Information Warehouse solution. The architecture is complemented by products and services available from IBM and non-IBM sources to implement that solution. In addition, FAA (see Chapter 4, "Financial Application Architecture" on page 33) is a vehicle for capturing finance industry-specific data entities and processes as well as a way of managing dependence of applications, including information systems on existing technology.

Part 3. The Technology View

Chapter 4. Financial Application Architecture	33
4.1 The Architecture Structure View	36
4.1.1 The Application Layer	37
4.1.1.1 End-Use Dimension	37
4.1.1.2 Data Dimension	40
4.1.2 The System Layer	41
4.2 The Enterprise Environment View	41
4.2.1 The Information Model	42
4.2.2 The Development Environment	45
4.2.3 The Network	46
4.3 Financial Services Data Model	46
Chapter 5. Information Warehouse Framework	49
5.1 Value of the Information Warehouse Framework	50
5.2 Why Data Replication	51
5.2.1 Operational Systems	51
5.2.2 Database Technology	52
5.2.3 Cost of Data Access	52
5.2.4 Historical Data	53
5.2.5 Ownership	53
5.2.6 Point-in-Time Data	53
5.2.7 Reconciliation	53
5.3 The Information Warehouse Architecture	53
5.4 Using the Information Warehouse Architecture	55
5.5 Access Enablers	56
5.5.1 Embedded SQL	58
5.5.2 SQL Call Level Interface	58
5.5.3 Distributed Relational Database Architecture	59
5.6 The Finance Industry	59
5.6.1 Information Warehouse Architecture Goals	60
5.6.2 Information Warehouse Architecture Focus Areas	60
5.7 Data Replication	61
5.7.1 Copy Tool Usage	61
5.7.2 Single Point of Control	61
5.7.3 Interface Orientation	62
Chapter 6. Finance Solution Thread Overview	65
6.1 Business Assumptions	66
6.2 The Solution Thread	67
6.3 The Business Function	69
6.4 Information Requirements	70
6.4.1 Customer Information	70
6.4.2 Profitability Analysis	72
6.5 Data Replication Strategy	72
6.5.1 Minimize Systems Administration Workload	73
6.5.2 Leverage Investment in Products and Strategy	73

6.5.3 Minimize Data Copying Cost	73
6.6 System Configuration	74
6.6.1 Platform Configuration	75
6.6.2 Communications Configuration	75
Chapter 7. Organization Asset Data	77
7.1 Foreign Currency and Traveler's Checks Model	79
7.1.1 Logical Data Model	80
7.2 The Entities	81
Chapter 8. Data Replication Tools	83
8.1 Data Replication Type Requirements	85
8.1.1 Business Requirement	85
8.1.2 Update Propagation	87
8.1.3 Copy Consistency	87
8.1.4 Update Sequence	88
8.2 Data Replication Technologies	90
8.2.1 Data Access Protocol	91
8.2.2 Update Propagation	91
8.2.3 Refresh Propagation	91
8.2.4 Archive	91
8.3 Data Replication Products	92
8.3.1 DataHub	92
8.3.2 DataPropagator Relational	92
8.3.2.1 Server Structure	95
8.3.2.2 DataPropagator Relational Capture Program	97
8.3.2.3 DataPropagator Relational Apply	100
8.3.2.4 DataPropagator Relational/2	101
8.3.2.5 Security and Audit	102
8.3.2.6 Pruning	102
8.3.2.7 Tuning and Control	103
8.3.2.8 External Sources	103
8.4 Implementing the Solution Thread	103
Chapter 9. Conclusions	105

Chapter 4. Financial Application Architecture

In Chapter 3, “Business Requirements” on page 23, we present a justification for using architectures to address strategic industry pressures. In this chapter, we investigate the FAA as an example of such an architecture. We also discuss the Financial Services Data Model as an underlying model to the architecture. That is, the model serves as a basis for managing the things and activities in the finance industry business that need to be administered by data processing systems. The overall goal is to respond to the pressures on the industry and minimize the cost of the information systems used thereby.

An architecture is a strategic approach to strategic problems

It is doubtful whether the finance industry would ever be able to reduce the cost of its information systems unless a broad set of information models and standards is accepted, by organizations and software vendors alike, as a basis for industry applications. IBM has developed the FAA as a comprehensive base for the finance industry. FAA consists of a set of software architecture guidelines, interfaces, methods, and tools for financial applications. IBM defines the FAA structure on three levels, as follows:

An architecture helps meet business pressures at minimal cost

- Conceptual** The conceptual level defines the scope of the architecture. It identifies the major components and describes the framework that permits the components to interrelate.
- Logical** The logical level defines the components’ functions and the semantics that they use, often identifying the components’ verb sets, methodologies, and behavioral conditions.
- Physical** The physical level details the syntactical constructs of the architecture. It establishes the message, data, and interface structure used in the architecture as well as the specific behavioral constraints of the components.

The FAA strategy has three key aspects, or views, as follows:

Architecture structure

The architecture structure view defines a set of functions for all application and system components.

Enterprise environment

The enterprise environment view defines an information model, a network, a development environment, and a run-time environment. This view focuses on the operation of the FAA in these environments.

Enterprise infrastructure

The enterprise infrastructure view provides strategies for application integration, coexistence and migration, delivery, distributed systems, security, and systems management. This

view focuses on the use of specific components and environments to implement the strategies.

Figure 3 on page 35 shows the structure of the FAA and the interfaces available through views.

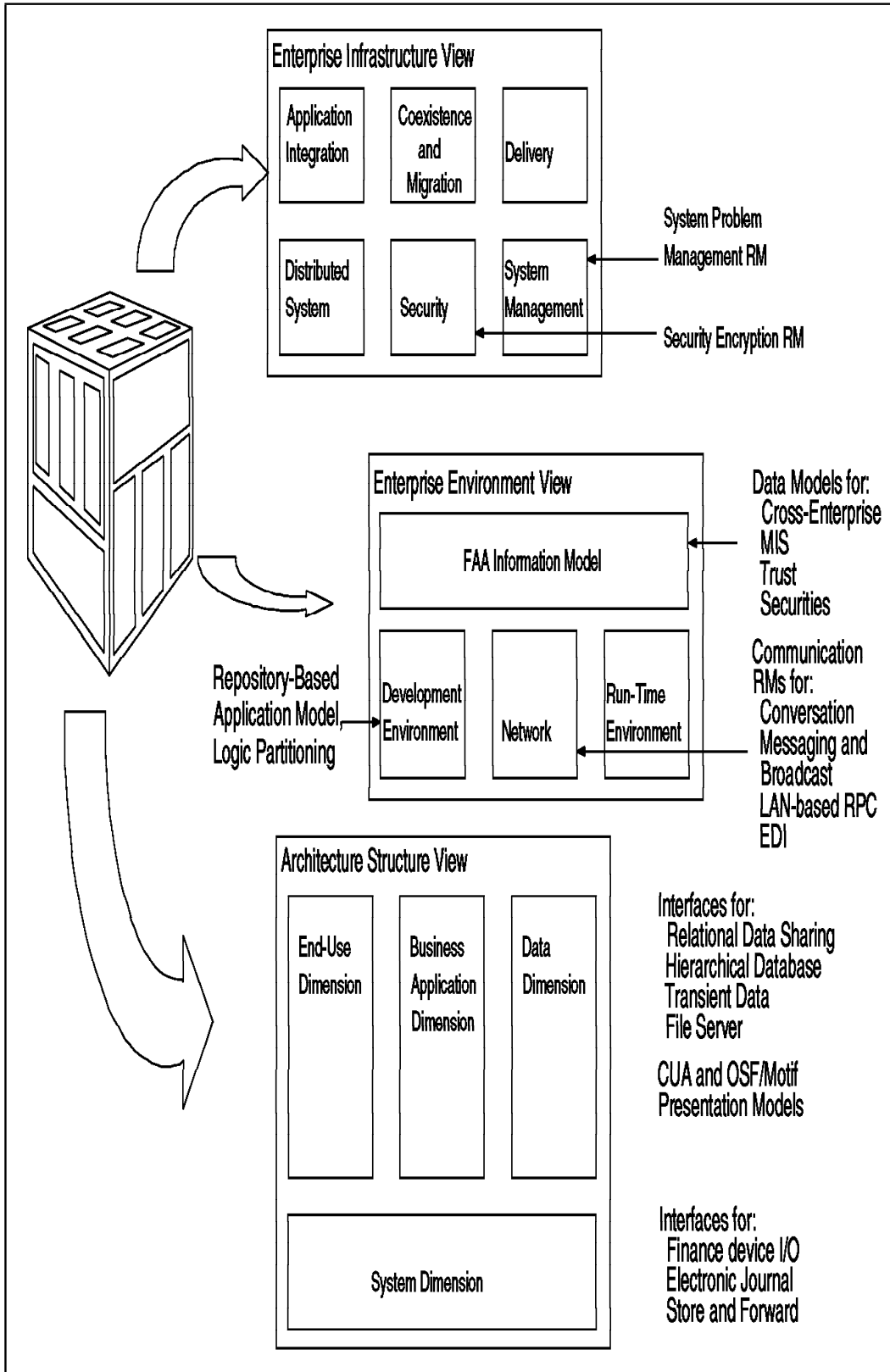


Figure 3. FAA Structure

4.1 The Architecture Structure View

Separate the business and data processing views

The architecture structure view consists of the application and system layers. The application layer represents business concepts as business objects. Business concepts so represented include accounts, customers, payments, transfers, and tellers. The system layer represents data processing concepts as data processing objects. Data processing concepts so represented include messages, entries in a database, and hardware devices. The purpose of the layers is to separate the business aspects of the system from the data processing aspects. Figure 4 shows the separation of the application and system layers.

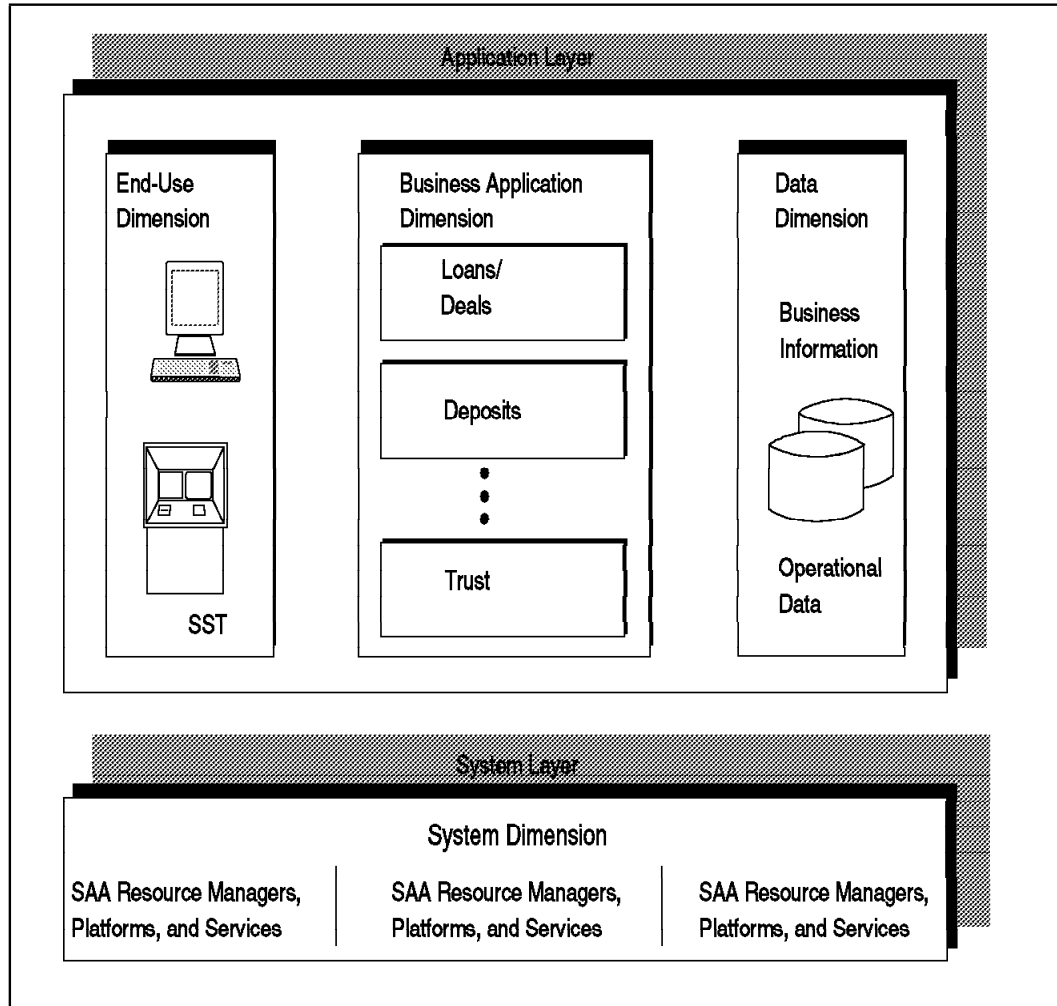


Figure 4. FAA: Application and System Layers

4.1.1 The Application Layer

Application layer components supply processing logic for the functions of an integrated financial system. These components isolate the application logic from the application composition, business procedures, and database designs. Any of the components of the application layer, including business action components, business services, and business object components, can be application layer components. The application layer is made up of the end-use and business application dimensions.

4.1.1.1 End-Use Dimension

The end-use dimension separates presentation logic from the business application. It provides for several different presentation styles, including an entry model, a graphical model, and an object-oriented (workplace) model. The end-use dimension also defines standard terminology, end-use components, interfaces, and rules, as well as graphical icons that financial service applications use. These are based on CUA and OSF/Motif** specifications, which are also found in the Information Warehouse architecture specifications—end-user interface for informational applications. The end-use dimension stresses consistency of user interfaces.

The end-user interface is data processing resource

This emphasis is consistent with one of the four focus areas of *Information Warehouse Architecture I*. Specifically, the *Informational Applications* focus area identifies four user interface environments as a standard across applications. The objective in the Information Warehouse architecture for informational applications is the same as that in FAA for business applications. In both cases, the architectures attempt to minimize the training necessary for end users by making applications look and feel the same.

Consistency of end-user interfaces is valuable

Business Application Dimension

The business application dimension provides guidance for integrating applications. The functions and services provided by general-purpose applications that conform to the FAA, such as office and informational applications, will be available to other FAA applications. The functions and services can be accessed through the programming interfaces of the general-purpose applications.

FAA integrates business applications

The Information Warehouse architecture takes a function approach to providing informational applications, rather than a product approach. The desktop suite of informational application functions is enabled by interfaces between the individual functions. This lets the end user start with any set of informational application functions and expand it over time. The function and interfaces orientation allows the functions to work together even as new ones are added to the desktop.

The Information Warehouse architecture integrates informational applications

Application partitioning increases software value

This dimension partitions applications into business logic components. The partitioning minimizes the effects of change, as business logic becomes less context sensitive. Each application function performs a well-contained financial activity and is not affected by changes outside its boundaries. This approach is similar to the object-oriented philosophy of methods. It promotes reuse of business application components. Figure 5 on page 39 shows the scope of the business application dimension. The headings represent the major families of financial products, often called the lines of business. The column on the left describes the major categories of financial business processing. A key strategy is that these applications contain application components that can be reused across several business functions, products, and services.

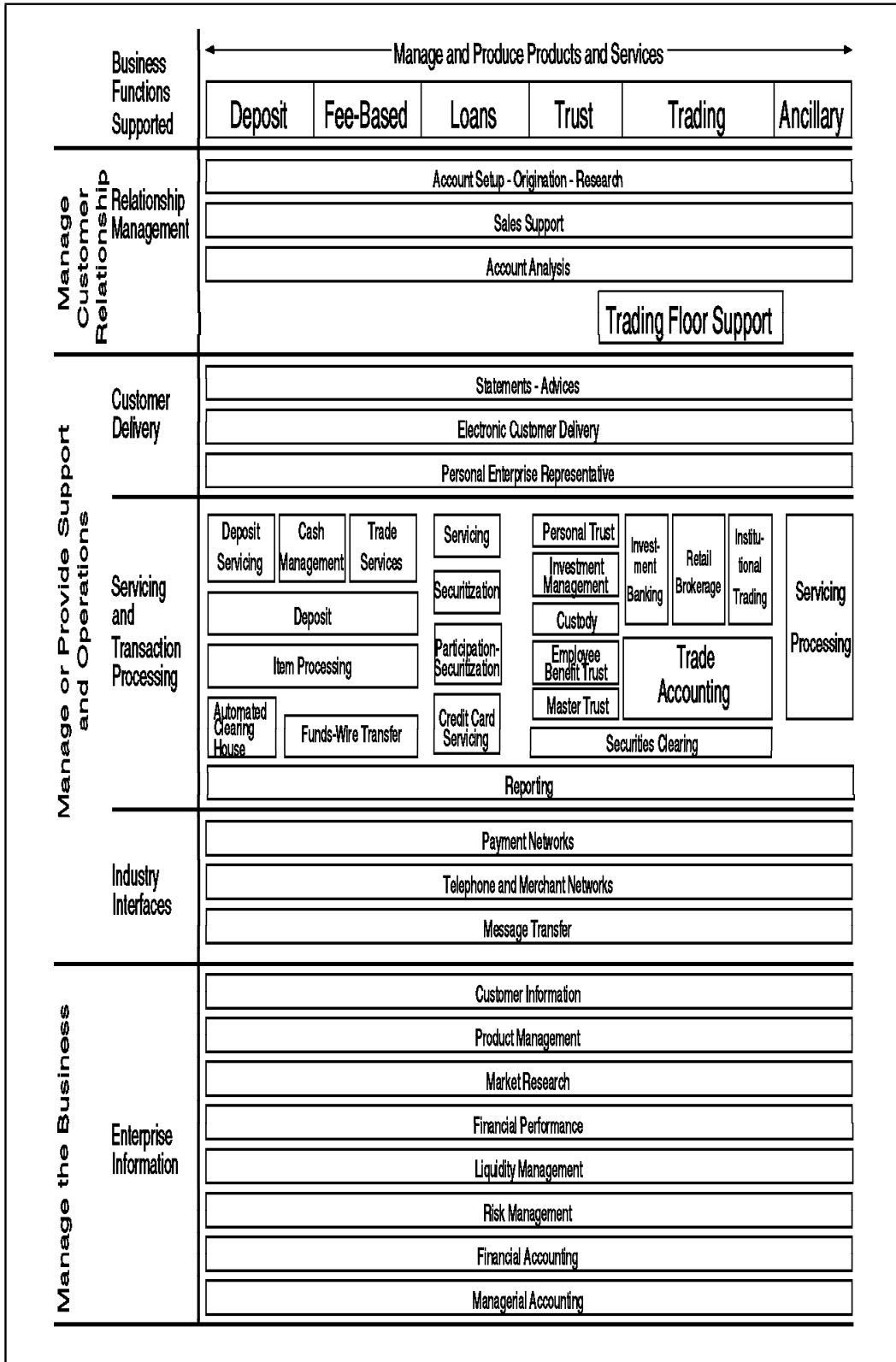


Figure 5. FAA Enterprise Application Framework

4.1.1.2 Data Dimension

The data dimension contains the logical view of the application data. The dimension uses data isolation techniques to screen the application from changes in database structure, data representation, or data technology. This dimension has direct parallels to the Information Warehouse architecture. It manages data access by isolating applications from the data itself. In Information Warehouse architecture terms, it is defining and managing the implementation of the *Access Enablers* component. Figure 6 on page 41 shows the separation of data available to application developers and users from execution services. This view is relational even though the data stores may be implemented in nonrelational data structures.

The data dimension components perform the following services:

- Manage data access
- Map the logical data definition to physical data management and storage systems
- Manage information distribution and integration.

FAA and IW architecture both modularize systems

The FAA data dimension incorporates two concepts integral to *Information Warehouse Architecture I*: the information dictionary and a single access method for users. The information dictionary in the FAA provides a function similar to that of the Information Catalog in the Information Warehouse architecture. It provides a short description of the data, the current status of the data, and a list of applications that can access the data. The single access method for users in the FAA is based on SQL for business applications just as it is in the Information Warehouse architecture for informational applications. Access to enterprise data is handled in the FAA by a translation mechanism that locates, extracts, reformats, and returns the data to the applications.

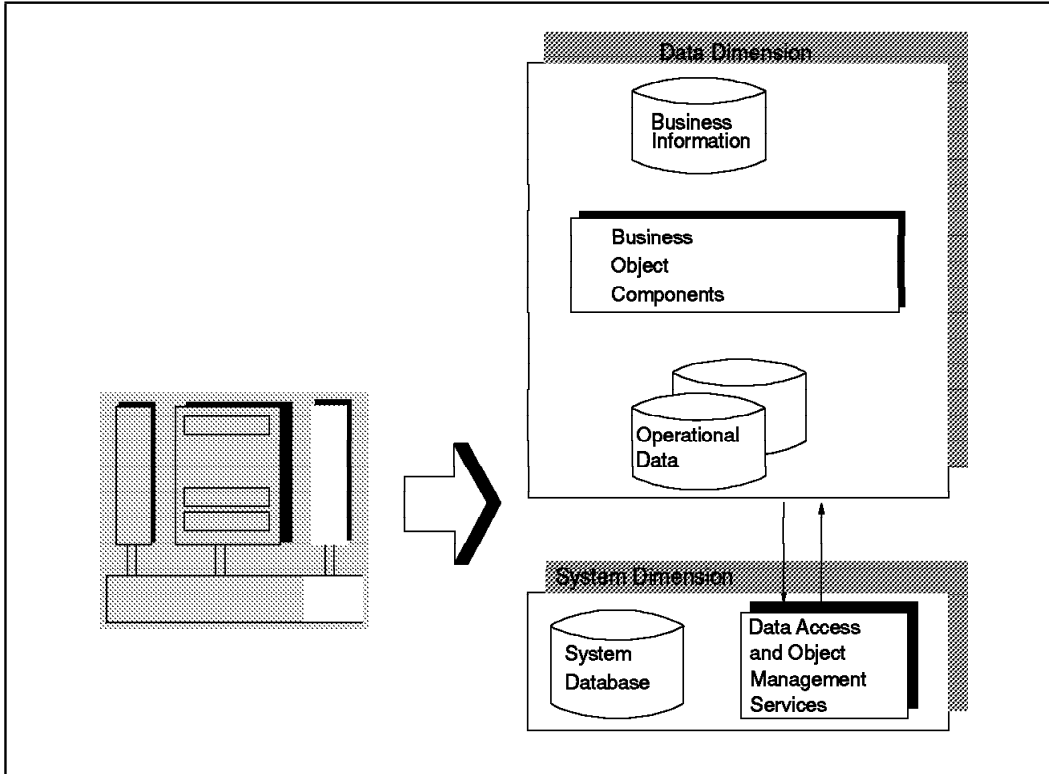


Figure 6. The Data Dimension

4.1.2 The System Layer

The FAA system dimension deals with the operating systems, subsystems, services, resource managers, and other facilities that support the deployment and operation of applications. The system dimension contains system control services, communication and common communication support, and Common User Access support.

4.2 The Enterprise Environment View

The enterprise environment contains an information model and defines the development, network, and runtime environments.

4.2.1 The Information Model

The information model bridges two views of data

The FAA information model is the point of transition between the business view of the enterprise and the information systems view. The model catalogs the information elements in the enterprise. It provides a mapping that enables financial personnel and information systems staff to communicate with each other. The FAA information model structures information resources into a single information image. The model consists of a set of submodels as shown in Figure 7.

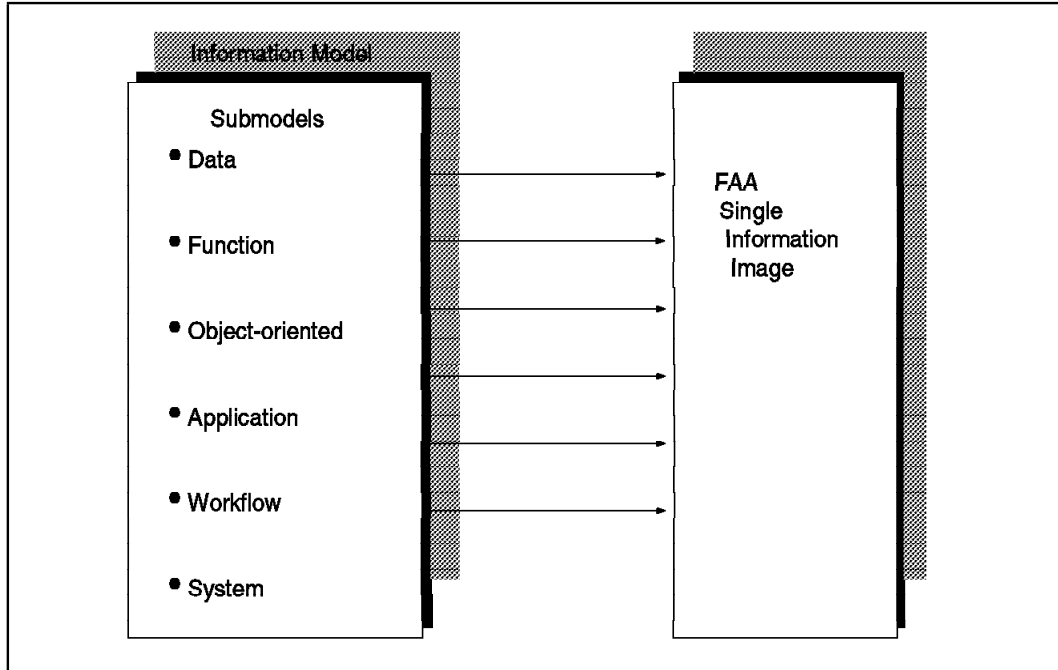


Figure 7. FAA Information Model

The FAA submodels each address a different area of business or technology function. These submodels provide the following functions:

Data The data submodel is a generic, global model of the structure of the finance enterprise's business data. It lets a broad range of applications use the same data throughout the enterprise.

Function The function submodel is a model of the valid operations that can be performed on data.

Object-oriented The object-oriented submodel represents sets of business information resources as objects. When an enterprise adopts the object-oriented programming, the enterprise uses this submodel in place of the data and function models.

Application The application submodel defines application packages. These application packages result from the combination of function and data components and additional input and output.

The application submodel provides the mapping of application components to their specific use.

Workflow The workflow submodel is a model of all of the activities involved in financial business operations. The workflows define the interaction among application packages. This submodel represents the roles of people, processes, and workflows needed to fulfill the mission of the enterprise. It provides the overall view of the business as processes.

System The system submodel represents system function and resources. It provides the mapping across location elements, system elements, and functions that these elements support. The system submodel translates the business organization into an information systems department.

While submodels may have data of differing types and physical structures, the information model has a uniform way to manage information resources. Finance enterprises use the information model to catalog information assets, communicate between business users and information system users, and enforce application consistency. The FAA data dimension aims at giving mappings and ensuring consistency between the business needs of the financial enterprise and the technological requirements of the information systems.

Each information submodel has conceptual, logical, and physical layers. The conceptual layer defines the scope and conceptual grouping of the submodel, the logical layer defines the reusable, generic structures of the submodel, and the physical layer defines the targeted physical environment of the submodel. Figure 8 on page 44 shows the relationship between the information model layers and its submodels. The submodels are the basis for the ongoing appearance of content model descriptions, model products, and applications that bring predefined implementations of the FAA structure to the finance industry. This strategy complements the Information Warehouse framework approach: that is, the Information Warehouse architecture evolves, products are introduced and are enhanced over time, and services help put the pieces together.

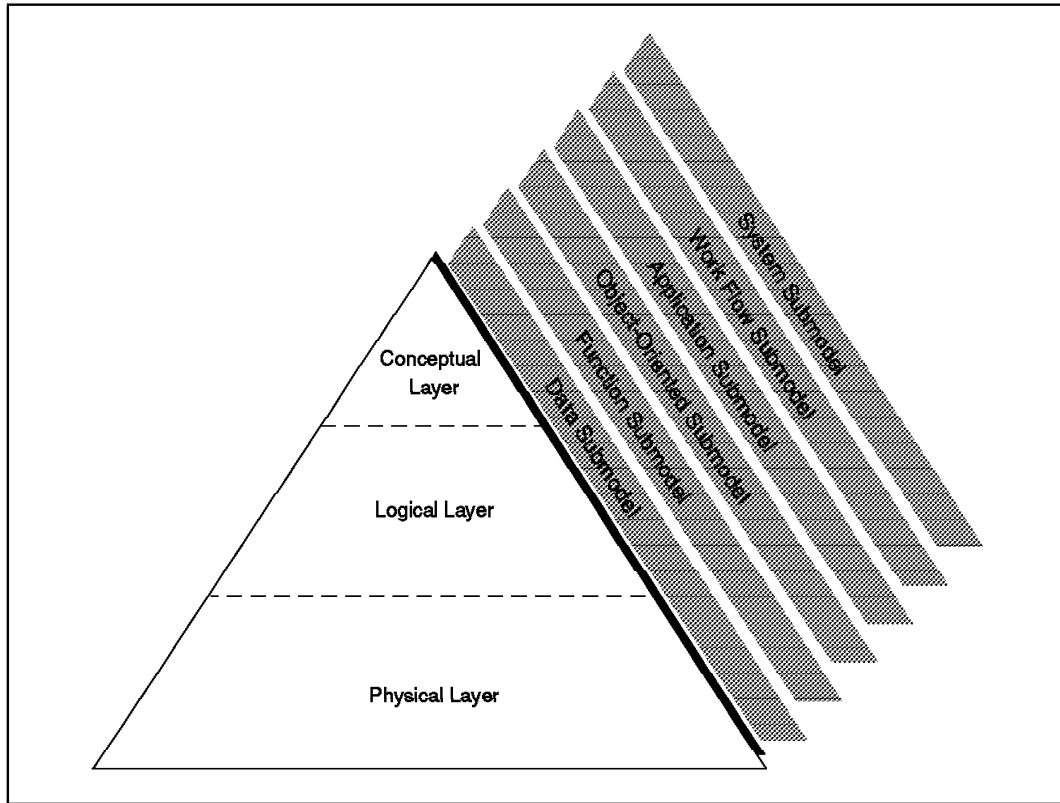


Figure 8. The FAA Submodels

The model bridges the business and technology views

Submodels represent either a portion or a view of an enterprise's information resources and interact to complete the enterprise view of the business. Table 2 on page 45 shows how information might be represented in the FAA information submodels. The key objective of the model is to supply the framework and structure to represent information having both business and technology attributes. Information can be seen in a business or technology dimension; the model provides the mapping between the two.

Table 2. FAA Submodels Implementation		
Submodel	Business Implementation	Information System Implementation
Data	<ul style="list-style-type: none"> • Customer name • Account balance • Credit history • Deals • Involved parties 	<ul style="list-style-type: none"> • Databases • Data structures • Data types
Function	<ul style="list-style-type: none"> • Amortization • Interest 	<ul style="list-style-type: none"> • Data operations • Calculations • Function logic
Object-oriented	<ul style="list-style-type: none"> • Customers • Accounts 	<ul style="list-style-type: none"> • Object-oriented implementations (data, function, inheritance, encapsulation)
Application	<ul style="list-style-type: none"> • New accounts • Deposits • Statements • Check processing • Letter of credit • Balance • Risk analysis 	<ul style="list-style-type: none"> • Executable modules • Production packages • Component linkages
Workflow	<ul style="list-style-type: none"> • Business operations manual • Cross-enterprise work activity 	<ul style="list-style-type: none"> • Application links • Multiple nodes, users, time spans • Manual operations
System	<ul style="list-style-type: none"> • Business organization • Users and responsibilities 	<ul style="list-style-type: none"> • Directives • Network management • Network topology • Function topology • System services

4.2.2 The Development Environment

The FAA application development environment includes the following:

- All components in the business application, end-use, and data dimensions
- Tools and services residing in the system dimension for processing these components
- Methodologies that support the use, development, and management of reusable components.

The FAA development environment has taken direction from the AD/Cycle family. This common direction will evolve and change according to the evolving content and success of AD/Cycle itself. The concepts incorporated into AD/Cycle describe well the application development process, at the life-

cycle level. There is work to be done, however, to bring the concepts down to a physical implementation level. A significant component of this challenge is the development of open interfaces by which work is passed from stage to stage and tool to tool within the AD/Cycle life cycle. FAA usability is therefore dependent on the progress and proliferation of AD/Cycle concept and product implementation.

Use the information model as a communications tool

The information model side of FAA is, however, a more established tool for use in the overall application development picture. The information model has an inherent value on its own as a high level of abstraction that can be implemented by a number of technologies and products. The high-level abstraction is in a form understandable by business analysts and executives. Lower levels of the model translate, step-by-step, the business analyst's terms into data processing terms usable for application development. This translation characteristic makes the model an ideal tool for improving communications between business and data processing professionals. All packaged information models require customization to be useful to any one enterprise. At the very least, the customization represents competitive advantage in the business object or activity that is being customized into the model. The FAA information model, and the objective of any packaged information model, is to represent a large enough subset of the business common to most enterprises in an industry to make it a sound business investment. IBM projects that the FAA information model covers upwards of 80% of the majority of finance enterprise's business needs.

4.2.3 The Network

IW data replication needs a network

Discussion of the network aspects of the FAA is beyond the scope of this publication. However, the network deserves attention as it is an important component of data replication in the Information Warehouse framework. Data replication is dependent on the underlying network topology and structure to copy data from the operational to the informational data store.

4.3 Financial Services Data Model

IBM has a comprehensive strategy for dealing with the issues confronting the industry. While a complete treatment of this strategy is beyond the scope of this publication, we do focus on one particular aspect of this strategy. In the discussion that follows, we limit ourselves to the role of information models in the finance industry; we ignore functional components of information technology such as transaction processing, accounting, network, and end-user systems and the application development environment.

A common way to represent organization asset data can contribute to achieving the objectives of integration and cost reduction. The common representation of the organization asset data normally takes the form of building an enterprise data model. The cost of producing an enterprise data model is high and, from a finance industry point of view, it is inefficient for multiple enterprises to expend resources building essentially the same enterprise data model.

The information model is key to integration

Rather, a core enterprise data model, available as a package for purchase, is more practical, with modifications possible to support competitive advantage of the individual enterprise. Few organizations have the resources to build an enterprise data model, and there is a reluctance in the industry to finance large projects with an uncertain return. This is particularly true where data processing projects are funded by end-user organizations. Information model development, conversely, is seen as a part of the information technology infrastructure, rather than an end-user application. An information model provides a base for integrated model-based application development.

A core model for the industry is optimal

The industry information model approach offers the credibility of the overall industry, rather than an individual enterprise organization. It transcends the individual organization to include the collective thoughts of industry experts, industry software vendors, and generalized software developers.

An industry approach broadens the perspective

IBM has developed the Financial Services Data Model (FSDM) in response to these business inhibitors. It models the data that might be encountered in the typical financial organization. FSDM can be customized to accurately represent a particular institution's information assets. It is a reference tool that can be used to define data requirements and for database design. If consistently used in all development projects, it can become the organizational reference for data standards. FSDM is an IBM offering that implements the data portion of the FAA information model.

FSDM is an industry-level approach

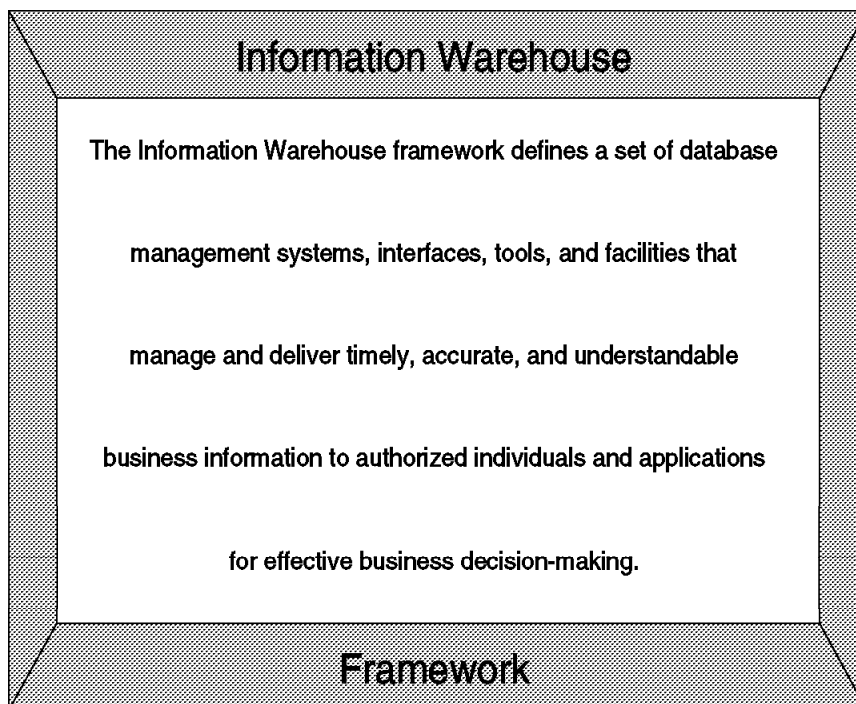
The initial promise of modeling and models has fallen short in realization for a variety of reasons. Products have failed to deliver on the promise of end-to-end modeling to application generation. Primarily, this failure is due to the inability of the individual products at each step of the application development life-cycle to talk to other products preceding or following them in that life-cycle. Technology developments can invalidate the premises on which models are built (for example, mainframe versus LAN implementation and relational versus object-oriented database). It could be argued here that the models themselves are not flexible enough to tolerate technology change.

Models and modeling are only a start

Despite these disappointments, there does not seem to be an alternative way to bring information together. If information is to be reconciled, there has to be a mapping between different operational and informational data. Enterprises individually and the industry collectively need to determine the level at which they should control data representation.

Models and modeling may be the only way

Chapter 5. Information Warehouse Framework



Enterprises have long recognized the opportunities that would be available if they could make better use of their data. Data is typically stored in many locations, in different formats, and managed by products from many different vendors. It is usually difficult to access and use the data across locations and vendor products. The Information Warehouse framework is a solution to this long-standing problem. IBM and other vendors are working to make access to data across vendor products and geographic locations easier by enabling their products to work together. The products and rules by which the products can work together constitute a framework. The Information Warehouse framework is designed to provide open access to data across vendor products and hardware platforms.

The framework: better use of enterprise data

IBM and vendor solutions fit into the framework

IBM and its business partners have been delivering database, data access, and decision support products which fit into the framework and make it possible for our customers to build effective integrated Information Warehouse solutions.

The framework enables access to all enterprise data

Enterprises benefit from the framework because they can increase the value of the investment that they have in current databases and files. Enterprises can get to the data that they need to effectively manage their businesses.

End users spend more time using data, less time finding it

The Information Warehouse solution includes decision support products. These applications can be used by end users to analyze and report business data from many parts of the enterprise. The Information Warehouse solution gives the end user access to data from other workstations, LANs, and host databases. The end users spend less time gathering and accessing the data, and more time in analysis and reporting of data.

5.1 Value of the Information Warehouse Framework

The Information Warehouse framework is a comprehensive solution having value beyond the simple collection of software products. The following characteristics differentiate the Information Warehouse solution from other approaches:

- **Published architecture**
Products that implement the published Information Warehouse architecture can work together with consistent user interfaces for easier operation. IBM and other leading software vendors have products that implement this architecture today.
- **Cross-platform coverage**
Database products reside on multiple platforms, and access is enabled to database products from a variety of software vendors on platforms from a variety of hardware vendors.
- **Architected connectivity**
Distributed relational database connectivity is architecture-based. IBM has products which support this architecture today.
- **Vendor support**
The Information Warehouse framework has the support of other vendors in the industry. These vendors are working with IBM to bring a wide variety of software solutions to market.
- **Architected systems management**
DataHub* is a cornerstone of the Information Warehouse solution. It is an architected solution for systems management, with the intent to cooperate with systems management products from other vendors.

- Integrated database tools

Tools for database design work with tools for application development, saving customers time in application modeling and development.

We can address the data delivery requirement to generate informational data from operational data by either iteratively writing applications to extract, enhance, and load information, or by using the architecture-based Information Warehouse approach.

5.2 Why Data Replication

There are two approaches to accessing data for informational analysis: accessing the operational data directly and accessing a replication of the data created by extraction, enhancement, and loading into the informational database. The consensus favors accessing informational copies of operational data rather than direct access of operational data. Some of the considerations leading to this preference are as follows:

- Operational systems
- Database technology
- Cost of data access
- Historical data
- Ownership
- Point-in-time data
- Reconciliation.

5.2.1 Operational Systems

Operational systems manage the day-to-day business activities. As such, they are critical to the ongoing viability of the enterprise. These systems often perform at the limit of the hardware and software with which they are implemented. They are often a key part of customer service, a major factor in the success of an individual retail enterprise in the very competitive retail industry. Therefore, the ongoing operation of the operational systems is the highest priority. The reasons for using copy-based informational systems with respect to protecting existing operational applications fall primarily into the areas of data accountability and application performance.

Operational systems operate at technology's limit

Data accountability includes security and audit considerations. Operational systems are designed from the beginning with a specific user community in mind. The data created and manipulated by those applications are carefully managed with respect to who has access to the data. The management is accomplished through operational policies and security function in the software. It includes allowing access of specific sets of data by specific users or by certain user group identifiers and associating specific users with those group identifiers. Specifying every combination of user and data is a considerable effort, so the group identifier approach is more practical. However, this may result in defining group identifiers for broad groups of users with diverse profiles to access operational data. At the very least, allowing informational access by knowledge workers would be an incremental burden on the security administrator. It could possibly be a major burden on the security software and a complication for the security policies of the enterprise. It

An isolated copy is less complicated to secure

is easier to have a copy of the data in an isolated informational environment where the access security can be handled independent of existing operational systems policies.

Data location impacts security

Data placement is also an issue in security. Copying data could increase the security exposure to the enterprise; once the copy is made, it is up to the possessor to manage security. However, controlled copying is more secure than allowing broad groups of users with diverse access profiles to access operational data. At the very least, fresh copies of data are controlled.

Performance of operational systems must be protected

Decision support activity is difficult to predict, but certain characteristics of this class of applications are well understood. Decision support queries tend to access large volumes of data; they tend to apply complicated, long-running manipulations against that data; and they may retain claims on that data for extended periods of human think time. The queries, then, can be expected to interfere with transaction systems because of extensive locking, heavy I/O activity, and high demands on CPU and buffer pool resources. Informational analysis against copies of the operational data rather than the operational data itself prevents this interference.

5.2.2 Database Technology

Database software and hardware technology favors copies

Mainframe database technology supports a wide range of operational function in terms of concurrent access and data transfer rate. The Information Warehouse architecture is platform-independent and is compatible with LAN-based operational databases and application strategies. The flexibility of the mainframe platform makes it an ideal location for data copies. The mainframe platform can support a wide range of data volume and user populations. It also leverages data by being a central location for data to be accessed across work groups or lines of business.

The LAN platform has value for certain informational uses

The LAN lags behind the mainframe in I/O interface technology and ability to process data in memory. Recoverability, availability, and security functions on the mainframe tend to have fuller function. This has historical reasons as well as reasons based on the sheer volume of data inherently found on the larger systems. There are, however, many situations where specific subsets of data can be processed effectively in the LAN environment.

5.2.3 Cost of Data Access

Bring the data copy to the knowledge worker

Communication costs are reduced and response time improved if a copy is created in one or more locations. The general rule is to bring copies of data as close to the knowledge worker as possible. The store structure in retail is an example of this strategy. The network topology includes LAN-based branches and central host databases. Frequently accessed data is accessed in a most cost-effective way on a LAN server rather than running queries on the host. Factors that influence placement of data copies on the LAN or the large server include the cost of data processing (host versus LAN server) and of sending the answer set from the host (LAN communication cost is lower).

5.2.4 Historical Data

Operational systems usually do not allow for historical data analysis, yet this is a major concern to business analysts.

5.2.5 Ownership

Legacy system history along with security and availability requirements have placed data remotely from business activity. Copying data allows for data placement closer to knowledge workers responsible for making decisions. Ownership implies identifying an individual who is responsible for the quality and currency of the informational object.

5.2.6 Point-in-Time Data

Operational data tends to change over time. A point-in-time picture of information may be necessary for comparisons and understanding trends. Point-in-time information is part of a historical database strategy.

5.2.7 Reconciliation

Reconciliation of operational data is impractical as a real-time operation. Staging of data and data replication techniques are necessary to perform the reconciliation required for informational analysis without impacting the operational environment.

5.3 The Information Warehouse Architecture

The primary goal of the Information Warehouse architecture is to define a basis giving end users and applications easy access to data. The Information Warehouse architecture (see Figure 9 on page 54) defines a structure, formats, protocols, and interfaces as the basis for implementing Information Warehouse solutions. This architected approach creates an environment wherein solutions are leveraged. The leveraging is realized by reusing individual component solutions across implementations and by integrating off-the-shelf components in those implementations. This is not to say that an Information Warehouse solution cannot be implemented without the architecture. Rather, the architecture contributes to the leveraging of effort and resources.

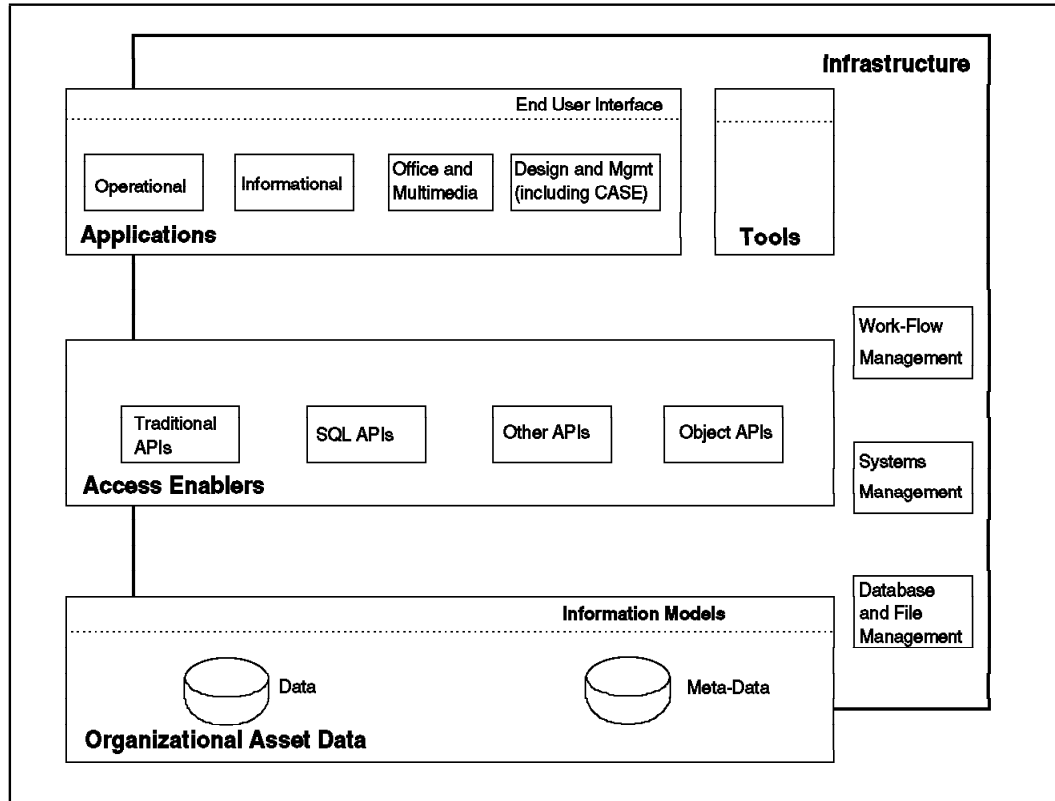


Figure 9. Information Warehouse Architecture

The long-term goal of the Information Warehouse framework is to provide access to data of all types in all stores in any environment, and the architecture is designed to accommodate the goal. The Information Warehouse architecture is open in that the interfaces are published and extensible in that software tools and data volume can be added without regressing the existing implementation.

The Information Warehouse architecture defines interfaces, protocols, and formats for accessing information in an Information Warehouse implementation. These interfaces are, as follows, grouped by focus area:

- Informational applications
 - End-user interface to informational applications
- Information Catalog and its access
 - Information Catalog API
 - Import/export interface to the Information Catalog
- Access to data
 - Embedded SQL API
 - Callable SQL API
 - Distributed Relational Database Architecture
- Data replication
 - Interface to the object handler meta-data
 - Tool invocation to tool interface
 - Interface to workflow management
 - Data staging interface.

The interfaces are identified and described in *Information Warehouse Architecture I* for the public domain. The Information Warehouse architecture

approach, using a component structure with interfaces defined between the components, and its openness regarding system platforms makes it easier for an enterprise to implement an Information Warehouse solution on its own or with the help of software vendors and service providers.

5.4 Using the Information Warehouse Architecture

Figure 10 on page 56 shows the three fundamental components of the Information Warehouse framework: the architecture, products, and services. The three components work together to build a foundation of an extensible, flexible, and scalable Information Warehouse implementation. The products and services are requirements, whereas the architecture is a recommended participant in the Information Warehouse framework strategy.

Information Warehouse framework for access to data

The products are considered requirements because the Information Warehouse framework is a software solution. The products component encompasses any software solution to the Information Warehouse framework function requirement, not just purchased software. Off-the-shelf software has its advantages in speed of implementation but costs real money and may require some investment in resources to customize and integrate into the enterprise's environment.

Off-the-shelf software for productivity

The services component refers to resources expended by people, be they enterprise knowledge administrators or personnel hired from outside the enterprise. In either case, people must do the work of designing, developing, and implementing software solutions.

Services: implementing the solution

The Information Warehouse architecture is a structured approach for building a solution. Information Warehouse implementations built without the architecture will solve a problem, but they may not be easily extended. The architecture-based, leveraged approach allows for reuse of software for similar functions across lines of business or specific requests. It allows the incremental addition of new function without disruption to the existing implementation. It also allows the growth in usage and data volumes without disruption of the existing implementation. The Information Warehouse architecture-based implementation is the recommended approach, though it is not the only approach.

The architecture for a flexible solution

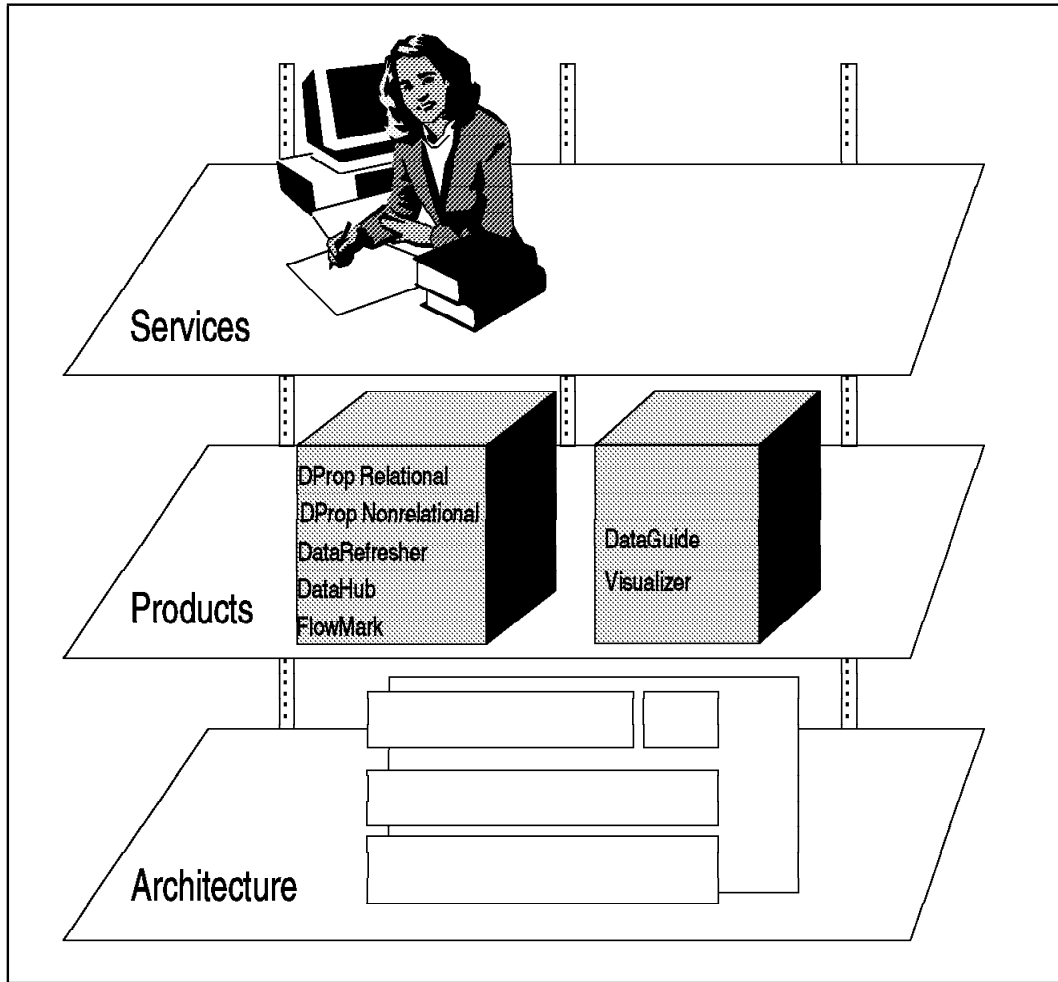


Figure 10. Information Warehouse Framework

5.5 Access Enablers

Access enablers connect applications and data

Access enablers is the layer in the Information Warehouse architecture between the application and the data (see Figure 11 on page 57). The data includes meta-data as well as real-time, changed, reconciled, and derived data. In the *Information Warehouse Architecture I*, the focus is on informational applications using SQL. These applications access relational databases locally with SQL or remotely with SQL and for example, DRDA. The value in using SQL and DRDA is that the application uses the same data access language to access local or remote relational databases. Nonrelational databases can be accessed by using SQL mappers in the implementation of the access enablers layer.

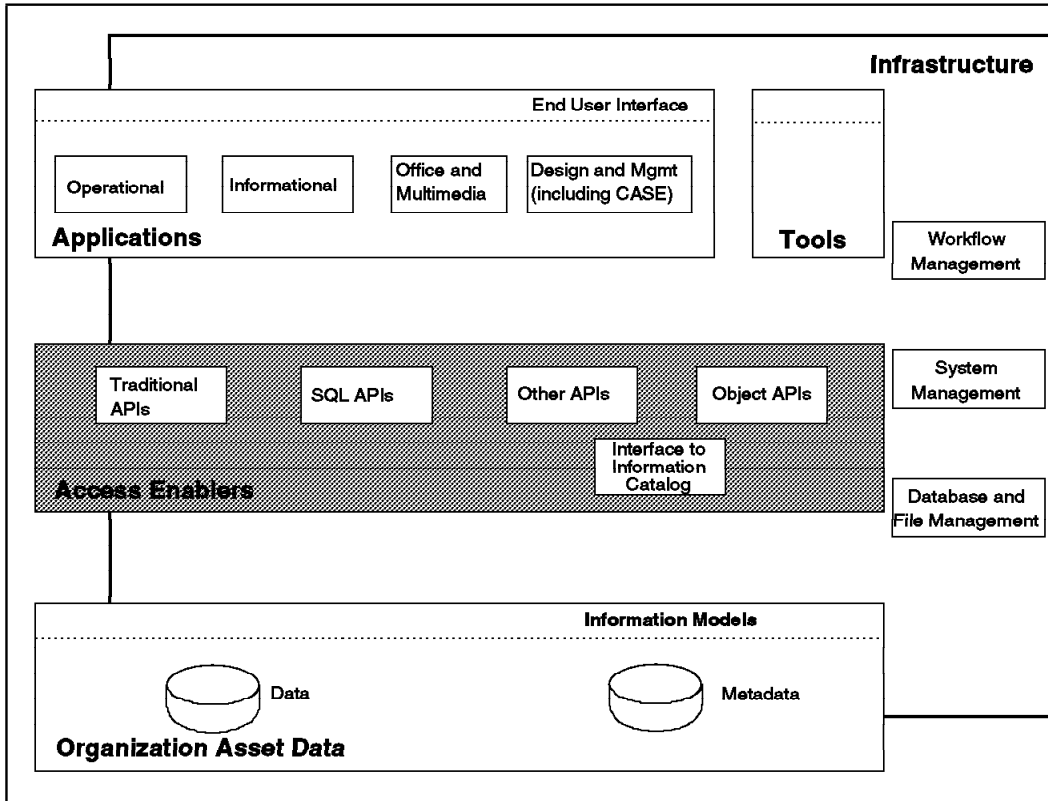


Figure 11. Access Enablers

The four focus areas of *Information Warehouse Architecture I* limit the discussion of the access enablers to the SQL application program interface and the interface to the Information Catalog. The concepts of *enabling* products and *deploying* products are key to understanding the use of the access enabler layer APIs and interfaces:

Interfaces are enabled, then deployed

Enabling An enabling product is a software program that accepts the commands defined in the API or interface and executes their defined function against a resource. DB2 is an enabling product for the SQL API, and the DataGuide products are enabling products for the interface to the Information Catalog.

Deploying A deploying product is a software program that submits requests for data resource in the form of commands defined in the API or interface. The commands are submitted to the enabling product for execution. Visualizer is a deploying product of the SQL API, and the DataGuide knowledge worker end-user interface is a deploying product for the interface to the Information Catalog.

An informational application could include commands defined in the interface to the Information Catalog and would become a deployer of both the interface to the Information Catalog and the SQL API. The advantage of this approach is that the knowledge worker continues to use the familiar environment of the informational application. The informational application would be enhanced by using business terminology stored in DataGuide.

SQL is used to access the informational and operational data categories and the interface to the Information Catalog is used to access meta-data. SQL is an industry standard data access language for performing relational operations, normally against data in a relational database. The access enablers layer also includes the definition of SQL mappers that allow SQL operations to be executed against nonrelational databases. Three key interfaces are included in the Access to Data focus area in *Information Warehouse Architecture I* and are related to the SQL data access language:

- Embedded SQL
- Callable SQL (commonly referred to as SQL call level interface)
- Distributed Relational Database Architecture.

The embedded SQL interface is an example of an interface taken from the public domain and is recognized by several standards bodies. The SQL call level interface is not as yet a standard; its prime focus is to enable software vendors to market shrink-wrap informational applications. The Distributed Relational Database Architecture was developed by IBM and is gaining recognition and support from a range of software vendors.

5.5.1 Embedded SQL

Embedded SQL refers to commands included in informational applications source code. The informational application must undergo special processing prior to normal compilation. It is this special preprocessing requirement that has fostered the development of the SQL call level interface.

5.5.2 SQL Call Level Interface

The SQL call level interface (CLI) is an alternative mechanism for invoking SQL from programs. The objective of CLI is to provide additional language commands (verbs) to extend the function of SQL. The most desired extension to SQL function is support of "shrink-wrap" applications. Software vendors would like to market applications utilizing the SQL but have experienced difficulty with the preprocessing and BIND requirements of embedded SQL. The informational applications are targeted for knowledge workers with little data processing skill. Requiring knowledge workers to go through the precompile, compile, and bind steps would diminish the acceptance of the informational applications by these users. The CLI introduces extensions to the embedded SQL command set which allow run-time precompile and BIND. The informational application can be used out of the box and does not require systems or database administration resource for the SQL portion of the application.

5.5.3 Distributed Relational Database Architecture

Distributed Relational Database Architecture (DRDA) is a communications vehicle for issuing SQL statements to a remote relational database and returning the results. The statements are executed against a remote relational database rather than a local database. Though there may be differences in the relational databases, such as the form and content of catalogs, the SQL statements themselves normally should not have to be changed to reflect the new location. It is an evolving architecture: DRDA level 2 introduces distributed two-phase commit protocols. IBM has developed and published the DRDA for use by software developers throughout the industry.

The overall objective of the access enablers component is to insulate the application from the enterprise data format and location. SQL mappers manage the mapping from SQL in the applications to nonrelational enterprise data when necessary. DRDA allows for specification of the location of the relational enterprise data. That is, an application using SQL can execute against a local database on that programmable workstation's DB2/2 database. That same application can execute against a relational database on the LAN server running DB2/2 by moving the database and causing the application to connect to DB2/2 on the LAN server. That same application can execute against a relational database on a remote server running any DB2 family member by moving the database, using DRDA and causing the application to connect to the DB2 database on the server.

5.6 The Finance Industry

The finance industry has made a huge investment in information technology. This investment is seen in the vast amounts of data and the huge collective portfolios of transaction processing and information systems. The finance industry has been deeply affected by the business networking phenomenon, with the associated impact on the industry's core logistics (see 2.3.1, "Business Networking" on page 17). The transformation to a business networking base forces a continuous focus on information systems. Despite the problems discussed in Chapter 2, "Finance Industry Perspective" on page 11 and doubts regarding return on investment, the industry continues to spend large sums of money on information technology.

IW is the key to data in the business network

The finance industry has always been a demanding user of information technology; its needs drive the development of new technologies. In particular, the finance industry has had a leading edge in high-transaction-rate and very-large-database processing. Today, the industry has concerns similar to other users of information technology. The data volume and network complexity create greater demands on the resources allocated to support them. These demands, specifically regarding access to enterprise data, are identified as typical across industries. The requirement to access enterprise data is the prime focus of the Information Warehouse architecture and is used as an introduction to the Information Warehouse framework discussion. The specific concerns of data access are as follows:

Finance's hunger for data access drives technology

- No single view of data
- Wide variety of user tools
- Lack of consistency

- Lack of useful historical capability
- Conflict between operational and informational needs of access
- Problems in administering data
- Proliferation of complex extract applications.

The problems of data access in the finance industry are a major concern, but are not unique to the finance industry. The problems are exacerbated by the high transaction rates and very large databases typical of the finance industry. Response time and transaction throughput alone preclude the use of operational databases for aggregate queries, making data replication a necessity. However, due to the amount of data involved, copying data can be costly, complex, and time-consuming.

5.6.1 Information Warehouse Architecture Goals

IW goals are finance's goals

The goals of the Information Warehouse framework include solutions to the data access problems experienced by most finance enterprises. The goals of the Information Warehouse architecture include the following:

- Provide information to end users about what data is available and how to access it using business terminology
- Offer a consistent programming interface—SQL—to formatted relational and nonrelational data
- Give location-transparent direct access to heterogeneous data
- Enable periodic extracts of heterogeneous data
- Facilitate creation of enhanced (reconciled or derived) data
- Enable update of reconciled and derived data through data changes
- Enable distribution of data to multiple locations
- Integrate the administration function.

5.6.2 Information Warehouse Architecture Focus Areas

Focus on a subset of the problem to be successful

Information Warehouse Architecture I has identified four focus areas, as follows:

- Informational applications
- Information Catalog
- Access to data
- Data replication

For more information on the four focus areas and their interfaces, see *Information Warehouse Architecture I*. The finance industry has needs for solutions in all four focus areas. We have chosen to discuss the implications of data replication in the finance industry. This does not preclude the other focus areas (covered in *Information Warehouse in the Insurance Industry* and *Information Warehouse in the Retail Industry*) from being relevant to the finance industry. It is simply a choice of where the initial attention and resources will be spent.

5.7 Data Replication

Information Warehouse Architecture I describes four typical data configurations, or collections of real-time, changed, reconciled, derived, and meta-data. An organization's data configurations may be composed of a variety of data stores (for example, relational and hierarchical databases and flat files) on a variety of dispersed systems. The configuration supports a methodology for organizing and categorizing enterprise data. The objective is to utilize reusable data replication software in an automated environment.

Data configuration and data replication are key to the solution

Creating and maintaining the reconciled and derived can be a complex task. It requires understanding the source and target data formats. It also requires a methodology for subsetting, cleansing, transforming, and transferring the data using copy tools. The role of data replication is to assist administrators in performing these tasks. Three of the fundamental premises of data replication, as they contribute to a technical strategy for the finance enterprise, are as follows:

Copy management assists knowledge administrators

- Copy tool usage
- Single point-of-control
- Interface orientation.

5.7.1 Copy Tool Usage

The copy-tool-based approach to providing informational systems is a technology issue. However, certain characteristics of the finance industry drive the use of specific copy tool function. Specifically, the finance industry has a strong need to perform informational application analysis on informational data that is consistent with the operational data. The single-point-of-control and the interface-based approach to a data replication solution are primarily technology-oriented requirements. Together, these strategies provide the maximum benefit to the business analyst community at a minimized cost to the information systems department.

5.7.2 Single Point of Control

The Information Warehouse architecture recognizes the need to manage heterogeneous databases in a heterogeneous hardware environment. Its data replication approach is based on the execution of data extract, enhancement, and load processes on any combination of hardware platforms. It would be inefficient for knowledge administrators to be trained on every platform to run the wide variety of processes. Therefore, the Information Warehouse architecture specifies a single, programmable-workstation-based point of control for managing data replication. Information Warehouse framework products deliver the single point of control through the interfaces for data replication; all server platforms are administered from the data replication administrator's programmable workstation. The data replication software delivers the software requests to the target platform and delivers the results to the data replication administrator and the target data platform.

The workstation controls the heterogeneous environment

DataHub is the basis for a single point of control

The control and user interface is provided by DataHub. This product enables a set of tool builder interfaces and services that ease tool integration to DataHub, reduce tool development effort, and enhance consistency across tools. The DataPropagator Relational and DataRefresher data replication products are enabled to DataHub, meaning that their functions can be invoked through DataHub. Figure 12 shows how DataHub provides a database network backbone for managing data replication. Other vendor products announced to run on DataHub include the following:

- OMEGAMON II** for DB2 (with a new DataHub/2 facility)
- PLATINUM Fast Load**, PLATINUM Fast Unload**, PLATINUM Rapid Reorg**, and PLATINUM Quick Copy**
- In2itive**
- BRIDGE/FASTLOAD**.

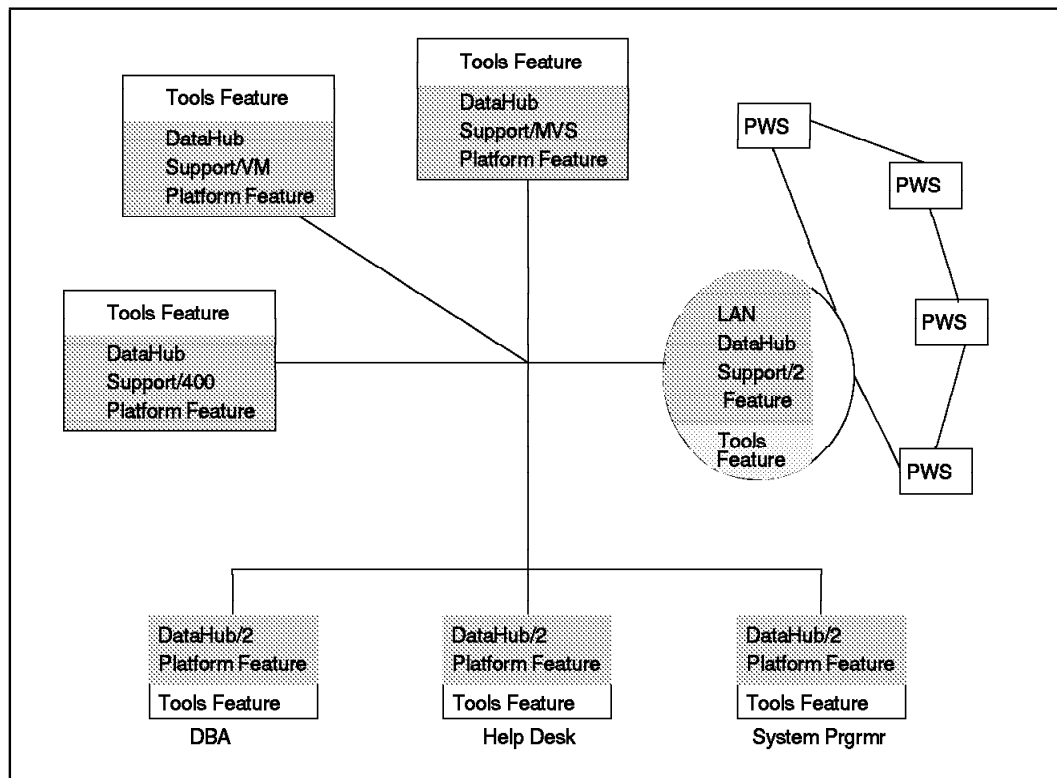


Figure 12. DataHub Environment

5.7.3 Interface Orientation

Interfaces allow problem solving, piece at a time

The complexity of data transformation and copying creates a challenge for implementing a data replication strategy. The sheer variety of hardware and software solutions for storing data in the (operational database) sources and (informational database) targets is challenging. The complexity of the data semantics and formats as well as the transformations required between the sources and targets add to the challenge. Data replication is made up of several components, each of which addresses a specific aspect of this challenge. These components are made up of a programmable-workstation-based administration and launch tool (DataHub), a workflow manager (FlowMark), and a suite of copy tools or data replication tools (for example,

DataPropagator NonRelational, DataPropagator Relational, DataRefresher, and DataHub copy functions). These components and the component solutions are integrated through the use of data replication interfaces to address the data replication requirement.

Information Warehouse Architecture I specifies four key interfaces for data replication, as follows:

Four interfaces are the bases for the solution

Object handler meta-data

The object handler interface is used to register objects, actions and tools. The interface is provided by the DataHub product. It is used at tool installation time and, in the case of objects, each time an object list is displayed for a tool.

Tool invocation

The tool invocation interface defines the data structures that are used for invocation time communication between a control point and a tool. The interface is provided by DataHub.

Workflow management

This is a drag-and-drop interface enabling easy movement of information between tool-specific copy requests into the workflow manager.

Data staging interface

This interface is used for passing data between tools using a common data representation format.

For more information on data replication, see *Information Warehouse Architecture I*.

Chapter 6. Finance Solution Thread Overview

We have discussed in some detail the overall business environment of the finance industry and the strategies that the industry enterprises have adopted for the 1990s. We have also examined the role information technology is likely to play in fulfilling the strategic objectives. The solution thread for the finance industry uses the Information Warehouse architecture and products to meet a strategic challenge in the finance industry. The solution thread is based on a common operational application designed to solve a tactical business problem: the management of the traveler's check business activity.

Tactical applications feed data to strategic decision support

The solution thread is presented in a "broad brush" manner so as to focus on the concepts of tactical (operational) applications, strategic (informational) applications, and the Information Warehouse technology that connects them. The solution thread illustrates the use of Information Warehouse architecture and products to solve a business problem.

IW is the technology for decision support software

The customer information system emerges as one of the key information systems. It is the vehicle for achieving a world-class customer relationship management capability. Among the requirements we cited were the following:

Business requirements translate into data processing requirements

- A work group (LAN) environment
- End-user, business-level guide to enterprise data
- Data replication function, including automation and update propagation
- Historical data capability for trend and segment analysis.

These requirements span specific hardware and software products as well as operational strategies for data management.

In this chapter, we present the case for environment and tools integration based upon a generalized, architected approach to meet information needs. The solution thread demonstrates how a customer relationship management objective can be helped by using the Information Warehouse architecture and deploying DataPropagator Relational, in conjunction with other programmable-workstation-based data replication products.

IW architecture is the integration point for the solution

6.1 Business Assumptions

We assume that the finance enterprise has adopted a business strategy with the following objectives:

- Expand customer base
- Increase ratio of valuable customers
- Increase marketing value of branch network
- Move to fee-based services.

Customer service and management contribute to profit

Customers are the finance enterprise's most important source of business both as depositors bringing cheap money and as borrowers representing revenue. The strategic objective of the finance enterprise is to expand the existing customer base. The existing and expanded customer base are maintained by a customer service strategy based on a world-class customer information system. The finance enterprise believes that only 20% of its customers contribute to the enterprises's profit. The remaining 80% either bring negligible profit or cost the bank money. Nonperforming accounts significantly increase the enterprise's cost of operation. The objective is to focus on the profitable customer base and concentrate marketing of additional services to that base.

The transaction's nature dictates its role in customer service

Branch infrastructure—particularly the ATM network—is a significant expense to the finance enterprise. The finance enterprise would like to leverage this investment as part of its customer service strategy. Most simple withdrawals and deposits are transacted through ATMs, eliminating these activities and the corresponding infrastructure component as a source for customer service improvement. The enterprise would like to use face-to-face contact to simultaneously market its products and services and increase consumer satisfaction. This preference points to the foreign currency and traveler's checks activity because it is a custom service and likely to require personal contact for the foreseeable future.

Compete on service, not the instruments in the service

Because of competitive pressures in the traditional interest-bearing activities—for example, loans—the financial enterprise is looking at alternative business activities to expand its revenue base. Fee-based services most easily meet that need. These services are implemented with generalized information systems that directly support customized service to each customer. The competition is then moved to the service rendered rather than the financial instrument used in the service. The finance enterprise that implements the best customer service information systems is the one that will succeed ahead of others.

The financial enterprise recognizes a need for a customer information system available to the branch personnel as a necessary tool in meeting the business strategy objectives. Customer information systems make it possible, during face-to-face interaction, to:

- Personalize the interaction based on customer history
- Identify best-fit products and services
- Project the profitability of a potential customer.

The financial enterprise is also interested in evaluating profitability of its foreign currency operation. To this end it wants to maintain history data based on extracts.

6.2 The Solution Thread

The solution thread is based on the Foreign Currency and Traveler's Checks application, which was developed as a sample application at the ITSO-San Jose Center. In the solution thread, we add a customer information system and make it available to the financial enterprise strategic planning staff. A full description of the original application can be found in *Client-Server Computing: The Design and Coding of a Business Application*. Readers are advised to consult this publication for the complete discussion of application development and implementation. The bank has a travel agency subsidiary called Finance Enterprise Travel (FE Travel).

A key system is built on an existing tactical application

The sample application uses a client-server model of computing that has become increasingly popular in all industries. Testimony to the growing interest in the finance industry for client-server computing is found in the newly formed Retail Bank Client-Server Consortium, organized by Stuart Research, a Cambridge, Massachusetts bank consultancy. For more information, see *Loaning Banks Some Courage*. The client-server model for computing helps the enterprise leverage the relative advantages of the LAN and mainframe environments.

Client-server computing is a key technology

The original application scenario contains sufficient detail for development of the solution thread. Some modifications to the original scenario are desirable to facilitate the role of an Information Warehouse solution in general and data replication in specific. Presentation of an Information Warehouse solution required expanding the data information dimension in the scenario. The application was primarily focused on the transaction itself to bring out the value of client-server computing. Also, the benefits of data replication could be better demonstrated by changing the placement of data in the network. We assume that foreign currency and traveler's checks are purchased in an over-the-counter, face-to-face transaction. The few minutes required to perform this transaction present the financial enterprise with an opportunity to attract a new customer and offer a fee-paying service. Figure 13 on page 68 shows the foreign currency transaction in the context of implementing the financial enterprise's business strategy.

Operational application → informational application

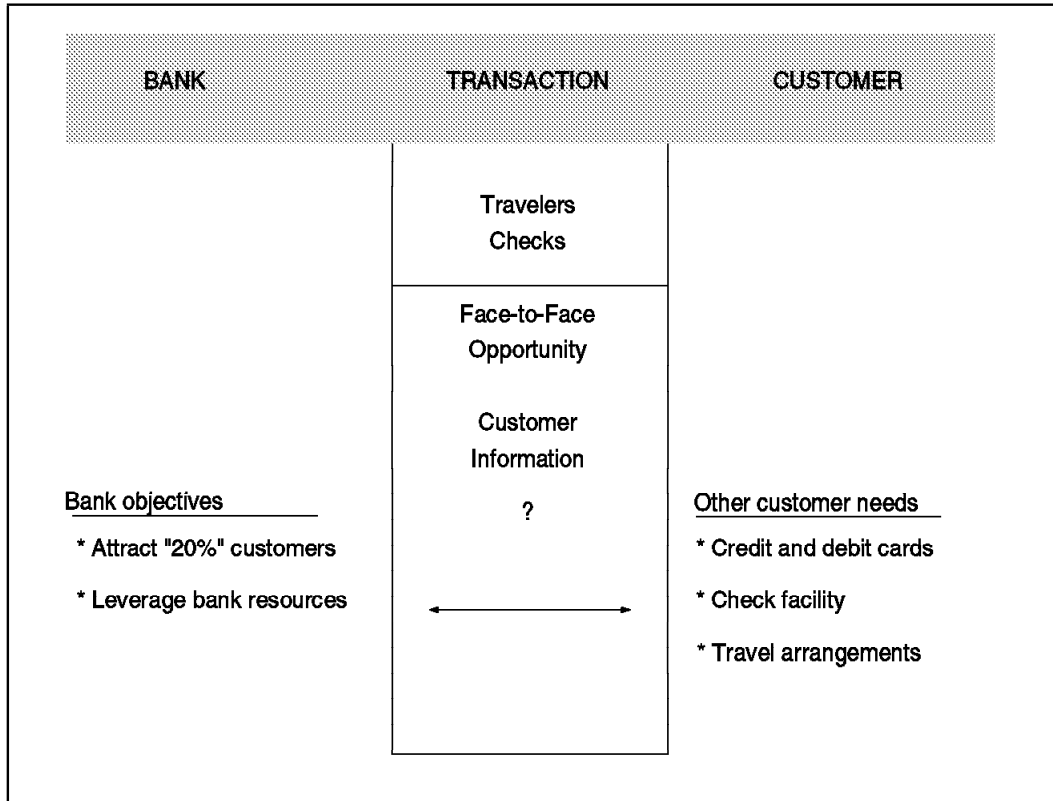


Figure 13. The Foreign Currency Transaction

Customer information personalizes customer service

The enterprise's counter clerk has two objectives in addition to selling currency and traveler's checks: establish rapport with the customer and offer additional services to the customer. The rapport is based on the clerk's familiarity with the customer's past dealings with the bank. The familiarity, of course, is based on the customer information system providing customer profile and transaction history information to the clerk. The profile in Figure 14 on page 69 is a two-tiered informational database, with a central copy residing on the mainframe and subsets propagated to the branches. Customer information thus assists personnel in establishing a more personal contact and providing world-class service.

Every customer contact offers a new business opportunity

The contact with the customer is an opportunity to sell other products and services. The clerk must have information about the products available across the enterprise's lines of business, information about the customer, and assistance from the information systems to match them. For example, the financial enterprise can offer in-country customers credit card, bill-paying, statement redirection, and home security services in addition to generating referrals to the FE Travel line of business. Extending services is based on the clerk's assessment with the assistance of the customer profile. Specialized software utilizing expert system technology might also be part of the evaluation.

A different set of services is relevant to the out-of-country customers. These customers are likely to be either on holiday or traveling for business, family, or educational reasons. Although they may not become long-term customers, they represent a short-term potential for some services locally and long-term potential for enterprise affiliates in the customer's country of origin. They may want, in fact, to use the bank's business travel services locally (for example, FAX and word processing facilities) or look to begin a relationship to be continued in the enterprise's affiliates in that home country.

Customize the offerings to the customer's situation

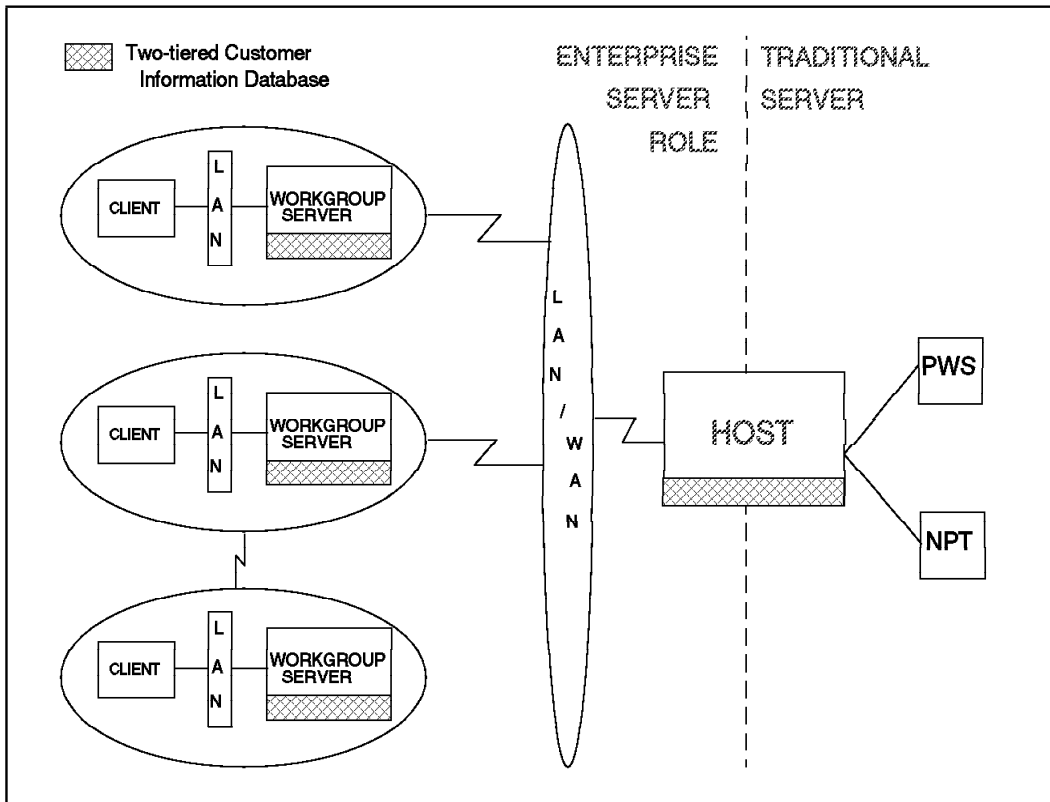


Figure 14. Two-tiered Customer Informational Database

6.3 The Business Function

The foreign currency and traveler's checks business is now described in the context of the finance enterprise's business. The banking business covers domestic and international operations. Domestic operations include all activities for domestic customers, including international trade. International operations are undertaken for both individual and corporate customers from other countries.

In the domestic operation, centralized departments provide head office functions. Branches serve as outlets to provide services to customers, using LAN-connected workstations and host connectivity. The foreign currency and traveler's checks application operates within this environment.

Providing foreign currency and traveler's checks is a large business with volume peaking in the tens of millions of dollars per day for each of the

major banks. The peak periods are the summer vacation months, though skiing holidays, Easter and Christmas breaks, and business travel lead to substantial transaction volumes year round.

Supply must be managed

Bank branches supply currency and checks on demand and try to maintain branch inventories of those financial instruments based on that demand. Branch inventory shortfalls in currency and checks are satisfied by ordering stock from the central department and mailing it to the customer or to the branch for collection. The branches also cash in traveler's checks and purchase currency.

Currency inventory affects profitability

The central department is responsible for maintaining appropriate stock levels centrally and at all branches. Maintaining these levels is very important for currency because stock holdings cannot accrue interest, and insufficient stocks mean lost sales. Central stocks are maintained by buying and selling on the foreign note market, through the currency dealers or from traveler's check suppliers (for example, American Express for U.S. dollars or Thomas Cook for UK sterling traveler's checks). Dealers maintain the exchange rates that the branches use on all their sales and purchases; these rates are applied immediately.

Each stock-holding branch must reconcile its stock against sales and purchases every day. The central department must also reconcile its stock against deliveries and dispatches each day. Branch profit arises from commission applied to each sale or purchase. The central department's profit is the favorable exchange rate applied to every sale or purchase.

6.4 Information Requirements

The key systems—customer information and profitability analysis—are potential platforms for using Information Warehouse architecture and products. In both systems, operational data is being generated and modified as part of the normal day-to-day business operations. The discussions of the key systems review the operational data as the source for informational data and the use of this informational data in an informational application environment. The discussions focus on the need to architect the operational and informational data

6.4.1 Customer Information

Complete database on mainframe server

Operational systems are the source for customer information. Data elements that contribute to both customer relationship management and to the more traditional analysis of the financial enterprise business performance include the following:

- Customer number
- Name and address
- Age
- Occupation
- Account numbers
- Loans if any
- Credit rating

- “Value” indicator
- Signatory to separate business or other accounts.

The credit rating is a weighted value representing profit potential to the bank based partially on the length of time the customer has been doing business at the finance enterprise. The credit analysis must be performed on a platform capable of storing the large history of data common to the finance industry. More current information such as a recent loan application may exist on a programmable-workstation-based system in a branch. Data replication is the key to bringing these various information sources together for informational analysis. DataPropagator Relational is a data replication tool that would support the replication of data to make it available for informational analysis.

Data replication plays a two-way role

Data replication also plays a part in distributing this informational data back to the remote business analysts who need it. The customer information database, in Information Warehouse terms, is a reconciled and derived copy of the operational data. To best illustrate the functions of DataPropagator Relational, we assume the informational data—the customer information—is stored in a relational database format, and we assume the relational database management system is DB2.

Information flows back to the source

Maintaining the informational data on the large server does raise the issue of access cost, availability, and performance. The financial enterprise wants this data to be deployed on the branch’s LAN. Clearly, copying the whole customer informational database to each of the branches is not practical or effective. As this is read-only data, the financial enterprise adopts an approach of copying subsets of customer information into each branch. A particular customer’s information is included in a branch’s subset if the customer opened an account or has recently executed a transaction there.

Place data for business reasons, not DP reasons

Transaction-based criteria for data distribution tend to produce a different customer informational database on a particular branch LAN over time. This raises the question of data accuracy if refresh is based on an update propagation. A customer executing transactions at multiple branches would appear in the customer informational database in each of those branches. The customer information for that customer may be inconsistent from branch to branch. The customer information application takes into account the two-tiered structure of the customer informational database. It first connects to a local database in search of customer information. If the customer information of interest is not there, it would connect to the large server.

The IW architecture spans platforms

The financial enterprise has a three-level data configuration, called configuration “D” in *Information Warehouse Architecture I*, made up of operational, reconciled, and derived data, with (probably overlapping) subsets at the branch level. This data organization presents significant challenges to the enterprise’s mission to manage enterprise data. Its viability, despite business need, hinges on data replication techniques and products to ease the burden of maintenance.

IW data configuration helps organize enterprise data

6.4.2 Profitability Analysis

IW architecture accommodates various data placements

Profitability analysis is the second key system for the finance industry. This section discusses the profitability analysis application in the Information Warehouse environment in a manner similar to the customer information system. We assume that the financial enterprise wants to track the commission received from its foreign currency and traveler's checks operation. The financial enterprise must maintain an audit trail of foreign currency and traveler's checks transactions to comply with legal requirements and assist government investigators. Customer order information is kept on the large server for this reason, then aged and archived from there. This data configuration is a departure from the original scenario as described in *Client/Server Computing: The Design and Coding of a Business Application*.

Use data replication for profitability analysis

The bank wants to understand commission aggregated by cashier and currency. An aggregate database is created, with a daily update. Each row may contain:

- Cashier identifier
- Currency
- Branch identifier
- Date
- Commission total
- Number of orders
- Local currency amount
- Foreign currency amount.

The foreign currency and traveler's checks application generates the raw transaction data to be used in the profit analysis application on the large server. Information Warehouse data replication techniques and products are used to copy, reconcile, and aggregate (derive) data to the large server. Information Warehouse Informational Applications are used to dynamically analyze and create reports or graphic presentations on commission. This information is used by both the central bank organization and the branches who have a need to understand their profit structure, customer demand, and staff's performance. Only business generated at the branch is kept at the branch's location in a three-level data configuration, with aggregate propagation.

6.5 Data Replication Strategy

Strategic decisions reduce long-term costs

The financial enterprise uses Information Warehouse architecture and data replication products to implement its solution to the key systems requirements. It expects to realize the following benefits from the architecture and products:

- Minimize systems administration workload
- Leverage investment in products and strategy
- Minimize data copying cost.

Information Warehouse architecture data replication strategies and products contribute to the realization of these objectives. The strategies are based on a component approach to the overall problem of copying data—with

enhancements—throughout the enterprise. The strategies are presented here as solutions to the objectives.

6.5.1 Minimize Systems Administration Workload

The data replication strategy focuses on a single point of control. This is crucial in a heterogeneous database management system environment. It would be inefficient for the data replication administrator to learn the systems management environments and tools on every platform. The better approach is to learn the environment of one platform and let the software on that platform control all other platforms. This reduces training expenses and increases the productivity of the copy administrator. We assume that these administrative tasks are executed on a LAN environment just as the business trend is toward this environment.

Single point of control for efficiency

6.5.2 Leverage Investment in Products and Strategy

The financial enterprise does not want to expend its own development funds on what it considers technology-oriented data replication tools. Rather, it wants to purchase software developed by vendors who are technically capable of high-quality solutions. The investment in these software products is protected by the Information Warehouse architecture and its open interfaces. The architecture is component-based; the information systems department knows that it can add or replace individual tools without threatening investments in other pieces of the overall solution. IBM has defined the interfaces to put the components together as part of the overall solution and publishes those interfaces for all vendors to use. The open interfaces enable the tools to be swapped or to be used together for slightly different requirements.

Leverage and protect the investment with the IW architecture

6.5.3 Minimize Data Copying Cost

Different business applications and different branches using the same application exhibit different rates and volumes of data change. The rate of data change is qualified by the frequency of update transactions and the number of rows or records updated. Ideally, we want to maintain informational data by propagating changed data when the number of rows or records changed is small. In situations where a large percentage of the data is changed, refresh is the preferred strategy. For business systems that are characterized by a high rate of updates and the percentage of rows or records changed is unpredictable, the decision to refresh or propagate is a technology decision. For more information on data replication, see *Delivering Data to the Information Warehouse*.

IW architecture responds to different data change activity

IW architecture provides flexibility to grow and change

Flexibility is an important requirement for the Information Warehouse strategy; the long-term nature of informational systems demands the flexibility to implement new and specialized tools as requirements change. The tools should be flexible and smart enough to choose between the update and refresh strategies. The Information Warehouse implementation must have individual tools specialized for different situations and some tools that are capable of functioning in multiple situations. The IBM solution includes both of these classes of tools. DataPropagator NonRelational supports propagation of a specialized source, IMS/DB. DataPropagator Relational includes the intelligence to perform well in either high- or low-update-rate environments.

Business needs drive data copy decisions

The need for a customer information system is a long-standing business requirement that has brought up difficult technology challenges. The financial enterprise customer database contains information on more than one million customers. The customer information system must be accurate to a business-specified point in time; the informational data cannot fall behind the operational data by more than two days. The cost of a full refresh strategy for the financial enterprise's hundreds of branches is seen as prohibitive, even if branches hold only subsets of the customer information data. Data change rate is estimated to be in the few percent range over two days of bank operation. This low rate of data change fits the profile of an update propagation application. The strategy is to capture changes to the master customer operational database as they occur and propagate them to the appropriate branch's LAN. Information Warehouse architecture and data replication tools make this scheme cost effective and very manageable from an administrative point of view.

DataPropagator Relational propagation is the answer

DataPropagator Relational manages change propagation through refresh and update propagation and is accessed from DataHub's action bar in the Action pull-down list. DataHub creates a programmable-workstation-based environment for database administration. The financial enterprise implements IBM's DataHub and DataPropagator Relational as the foundation of its customer information system.

6.6 System Configuration

The foreign currency and traveler's checks application runs on a LAN-attached OS/2 workstation that is linked to an MVS/ESA host. The original application presented in *Client-Server Computing: The Design and Coding of a Business Application* used the following system and database software:

- CICS/ESA
- DB2
- CICS OS/2
- DDCS/2
- Extended Services Database Manager.

The following new products are introduced in the Information Warehouse extension to the original scenario:

- DataPropagator Relational
- DataHub

- DB2/2 (replaces Extended Services Database Manager).

6.6.1 Platform Configuration

Figure 15 shows a simplified view of the hardware configuration used. The workstations on the LAN can be either client machines or local servers. They are connected by a communications controller to the host, which acts as an enterprise server. The hardware configuration corresponds roughly to the structure of the business described above. The functions provided by the enterprise server can be mapped to the functions provided by the head office. The functions provided locally by the LAN in each branch can be mapped to the functions provided by the branch. Each workstation on the LAN corresponds to the workstation that an employee of the branch (for example, an analyst or cashier) uses.

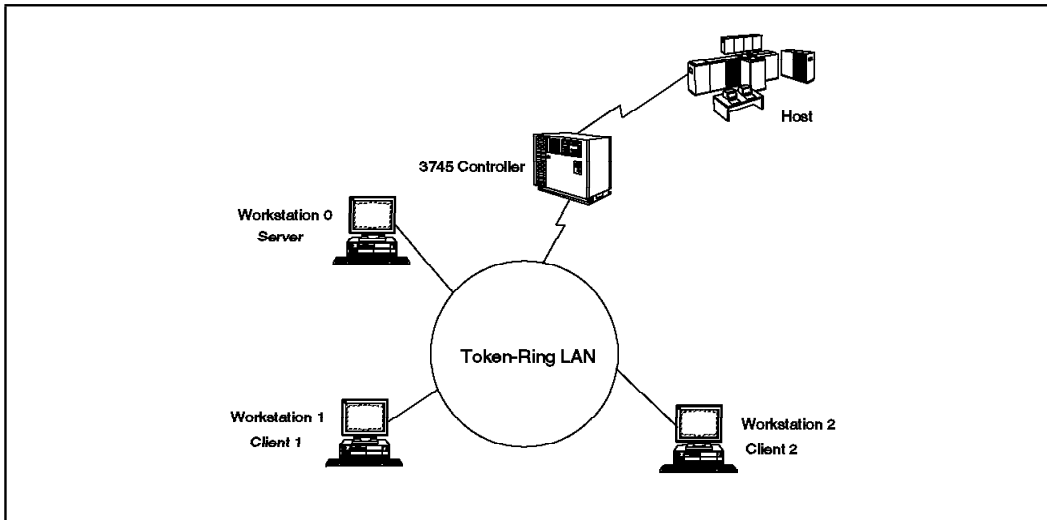


Figure 15. Hardware Configuration

6.6.2 Communications Configuration

Communications between the large server and LANs use the LU 6.2 protocol. The underlying communications hardware must therefore be able to supporting that protocol. The major software components in this scenario that use LU 6.2 communications are the following:

- CICS/ESA and CICS OS/2 for interconnectivity
- DDCS/2
- DataPropagator Relational (uses DDCS/2)
- DataHub/2 (uses DDCS/2).

DDCS/2 implements DRDA, which in turn uses LU 6.2. The DDCS/2 product facilitates a direct connectivity between DB2/2 and DRDA-compliant database servers. In this case, DDCS/2 connects DB2/2 to DB2 for MVS on the large server. Three workstations are shown to illustrate the connection possibilities. Additional client machines can be added and configured like client 1 or client 2, depending on the communication requirements of the newly added client.

Chapter 7. Organization Asset Data

Enterprises consider their data in all its forms to be a vital asset. For historical and organizational reasons, very few enterprises have a “master plan” for managing their data. Data exists in databases and files and is identified only as data objects by its data processing technology name. There is little categorization of the data objects, no enterprise-level directory telling to whom the data belongs, what the data means from a business point of view, or what role it plays in the applications that manipulate it. DataGuide/2—the product implementation of the Information Catalog—is a facility for describing what informational data objects exist and what they mean from a business point of view.

IW architecture brings order to the data chaos

The Information Warehouse architecture defines five categories of data with respect to how data is used by applications and specifies four configurations, or collections of the five categories. The categorization and configurations can be considered a data architecture. The categorization and configurations are a first step toward developing a data management system for the enterprise. Information Warehouse architecture’s *Organization Asset Data* component incorporates the categorization and configuration methodology. The data categories are as follows:

Data architecture facilitates data management

- Real-time
- Changed
- Reconciled
- Derived
- Meta-data.

Real-time data is created and manipulated by operational applications to run the day-to-day business. Changed data represents the changes that transactions make to the operational data. Reconciled data is an informational copy of the operational data with basic conversions of codes and resolution of inconsistencies of data stored in different parts of the enterprise. Derived data is an aggregated version of the reconciled data. Meta-data is descriptive data about the data in the other categories. It is used by knowledge workers searching for and trying to understand the data in those other categories. The meta-data is maintained in the Information Catalog. Figure 16 on page 78 highlights organization asset data in the Information Warehouse architecture.

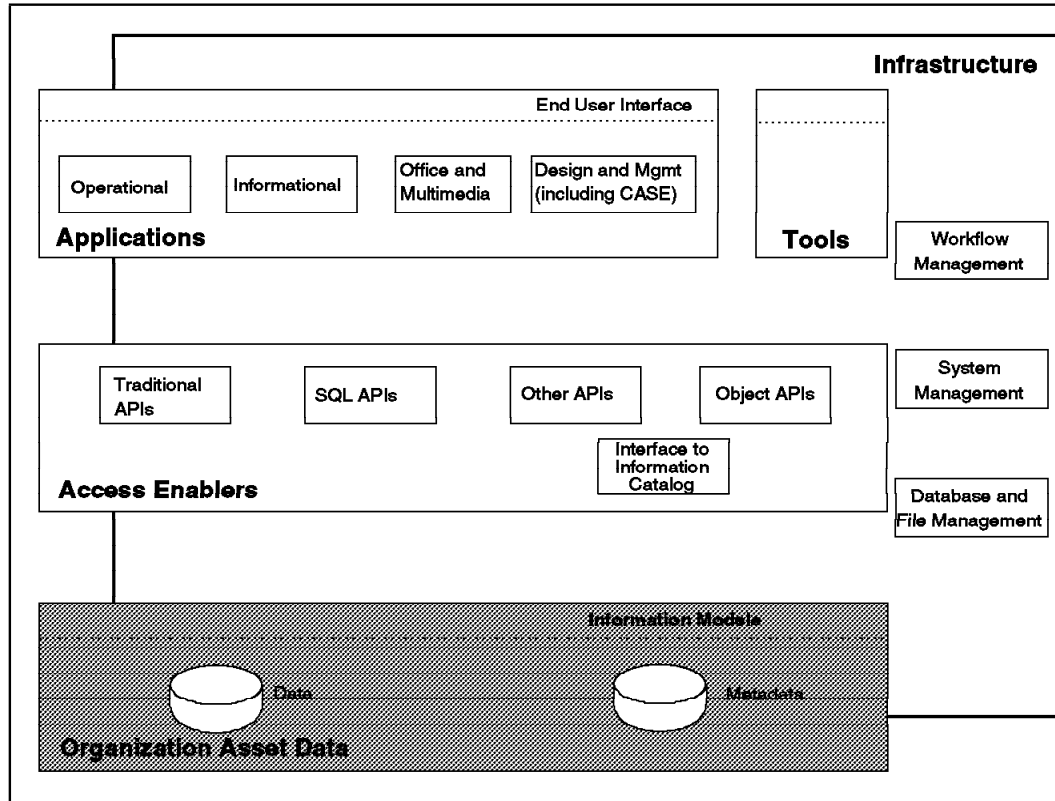


Figure 16. Organization Asset Data

An industry model presents the business view

The organization asset data is composed of two elements: data—real-time, changed data, reconciled, and derived data—and meta-data. The business view of the data is achieved through the modeling process, whether it is informal or tool-based. The discussion in Appendix A, “Models and Modeling” on page 109 lays the groundwork for that view. The business analyst understands the enterprise’s business and the business objects and activities that contribute to that business. A model is a translation of that view to the data processing view of the data processing objects and processes that mirror the business objects and activities.

The OAD categories complement the industry model

A model is a way of managing the categories of data defined in *Information Warehouse Architecture I* and creating a communication path between the business analyst and information systems department staff. The model starts from a business perspective and progresses toward the implementation of an equivalent data processing perspective. The data categories in the Information Warehouse architecture are geared toward how the data is used by applications. The Information Warehouse architecture view helps to make the data available to informational applications from the operational source from which it is extracted. The business model view provides the business semantics of the data. Both the business and the data processing technology views are necessary to maintain an understanding and management control of the data.

7.1 Foreign Currency and Traveler's Checks Model

We now look at the Foreign Currency and Traveler's Checks business and develop a model for it as an example of the modeling process, the model behind the process, and the relationship to the Information Warehouse. We keep in mind that this is a small piece of the enterprise's business and fits into the *logical model* level defined in both the FAA model discipline and the application development modeling discipline in general. The products of the modeling process are as follows:

- 1. Enterprise view** Industry model of the entire enterprise's business
- 2. Functional view** Logical model, a subset taken from the industry model that describes the Foreign Currency and Traveler's Checks business
- 3. Operational objects set**
The set of data processing elements that correspond to the logical model, which, in turn, is a subset of the industry model.
- 4. Organization asset data objects set**
The categorization of the data elements identified in the operational view into operational and informational data for the purposes of defining processes to create informational data from operational data.

Industry models, in general, focus on modeling the operational side of the business. This focus is a result of the historical emphasis on the development of operational applications and the ad hoc nature of informational data. The operational data drawn from the operational view corresponds to operational data in the Information Warehouse architecture categories. If an industry model includes informational business objects, then "operational data" drawn from the operational view is the source for informational data as defined in the Information Warehouse architecture categories.

Finance industry models rarely if ever include business report objects such as "What if we can lower the interest we pay on deposits by 10%?" This report—an informational object—is excluded from the operational-system-oriented model. Informational objects in a model tend to arise out of informational analysis of operational data. The informational object definition is fed back to the industry model, rather than being taken from the industry model.

7.1.1 Logical Data Model

Figure 17 shows an overview of the main data entities of the business and the relationships between the entities. It utilizes basic modeling constructs—boxes and arrows—to focus on the ends rather than the means of modeling. The arrows in this model indicate how two business objects—modeled as model boxes—relate to each other and the “sense” or direction of that relationship; for example, A→B means A owns B (not B owns A).

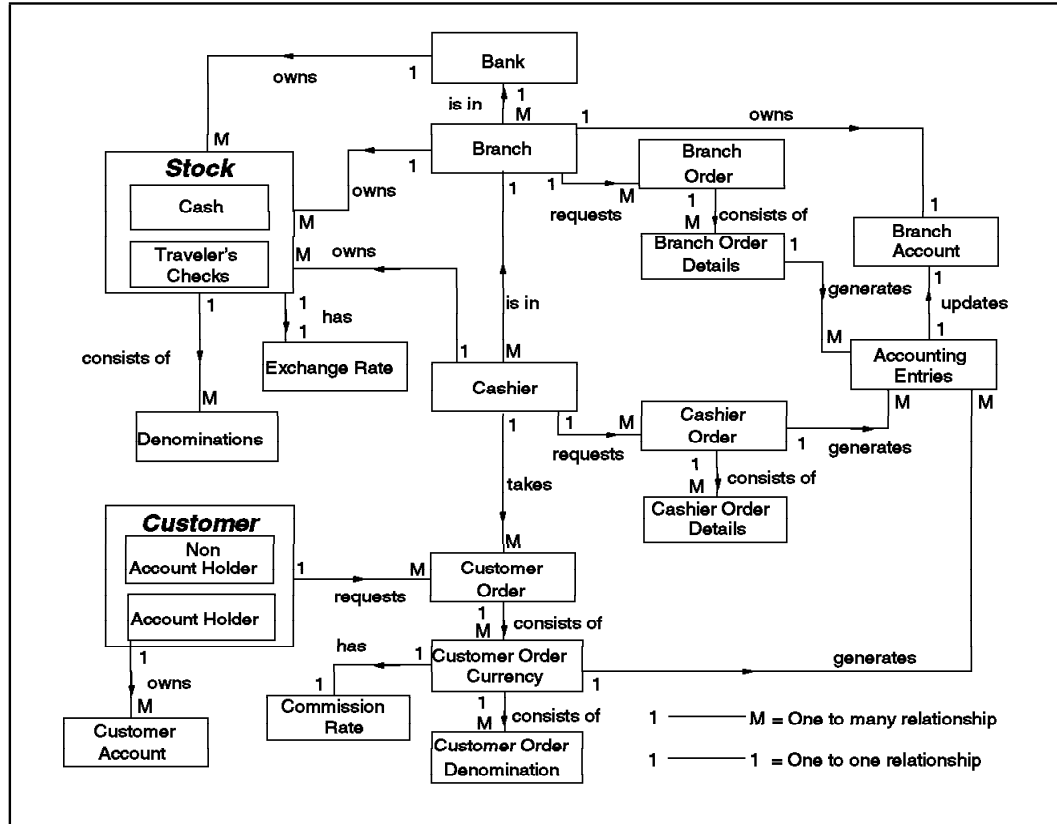


Figure 17. Business Data Model

The model describes the business

The data model shows that a cashier belongs to a branch, which in turn belongs to the bank. The bank, branch, and cashier have their own stock level. The stock consists of foreign currencies and traveler’s checks, which can be in different denominations. Each foreign currency has an exchange rate. The customer can be an account holder or a nonaccount holder in the bank. The customer owns at least one customer account if the customer is an account holder. A customer requests an order via a cashier. A customer’s order can consist of different foreign currencies in different denominations. A commission rate is charged for each currency ordered. A cashier and a branch can each place an order to replenish their stock level. All orders generate accounting entries, which update the branch account.

We now focus on the entities and their corresponding objects, as we progress toward the Information Warehouse data categories.

7.2 The Entities

Entities are the modeling term for business objects. We have described the Foreign Currency and Traveler's Checks business by walking through the model. The entities used in that walkthrough are the next step toward Information Warehouse data objects and the building blocks of the key systems. These entities are as follows:

- ORDER** The order contract is a piece of paper on which the agreement between the customer and the bank is described. The contract can be either a purchase or a sale, but not both.
- CUSTOMER** The customer is defined as the purchaser or seller of the stock. The customer may or may not be an account holder at the bank.
- CASHIER** The cashier is the employee of the bank interacting with the customer and the manager of the stock.
- STOCK** The stock is the set of currencies and checks held.
- BANK** The bank decides which branch can hold which stock at which levels. The bank replenishes branch stocks and takes excess stock from the branch. The bank also sets the rates and charges to be used by the branches.
- BRANCH** The branch decides which cashier can hold which stock and at which levels. The branch can enable a cashier to use stocks from another cashier at that branch.

Table 3 on page 82 shows the data placement. Note that the location of some data has changed with respect to the original scenario.

Table 3. Data Placement	
Data	Location
Currency	Client machine
Currency_denom	Client machine
Backup_currency	Local server
Backup_curr_denom	Local server
Exchange_rate	Host
Customer_order	Host*†
Cust_ord_denom	Host*†
Cust_ord_curr	Host*†
Branch_order	Host
Branch_ord_detail	Local server and host
Cashier_order	Local server
Cashier_ord_detail	Local server
Branch	Local server and host
Cashier	Client machine
Control_info	Local server
Commission	Client machine, local server, and host
Customer	Local server*
Legend:	
* Table location changed from the original scenario described in <i>Client-Server Computing: The Design and Coding of a Business Application</i>	
† Originally located at local server	

Chapter 8. Data Replication Tools

Data replication tools (also known as copy tools) represent to a wide range of software functions used to manage the copying, enhancement, and loading of data from operational to informational data stores. These tools make up one piece of the data replication strategy, which has as its overall objective the automation of copy processes. Automation of replication processes, in turn, is the key to making an Information Warehouse implementation feasible and reliable as a long-term component of the enterprise's information systems. Data replication tools are included in the Information Warehouse architecture's Tools component. Figure 18 highlights the Tools component in the Information Warehouse architecture.

Data replication is the key to reliable informational systems

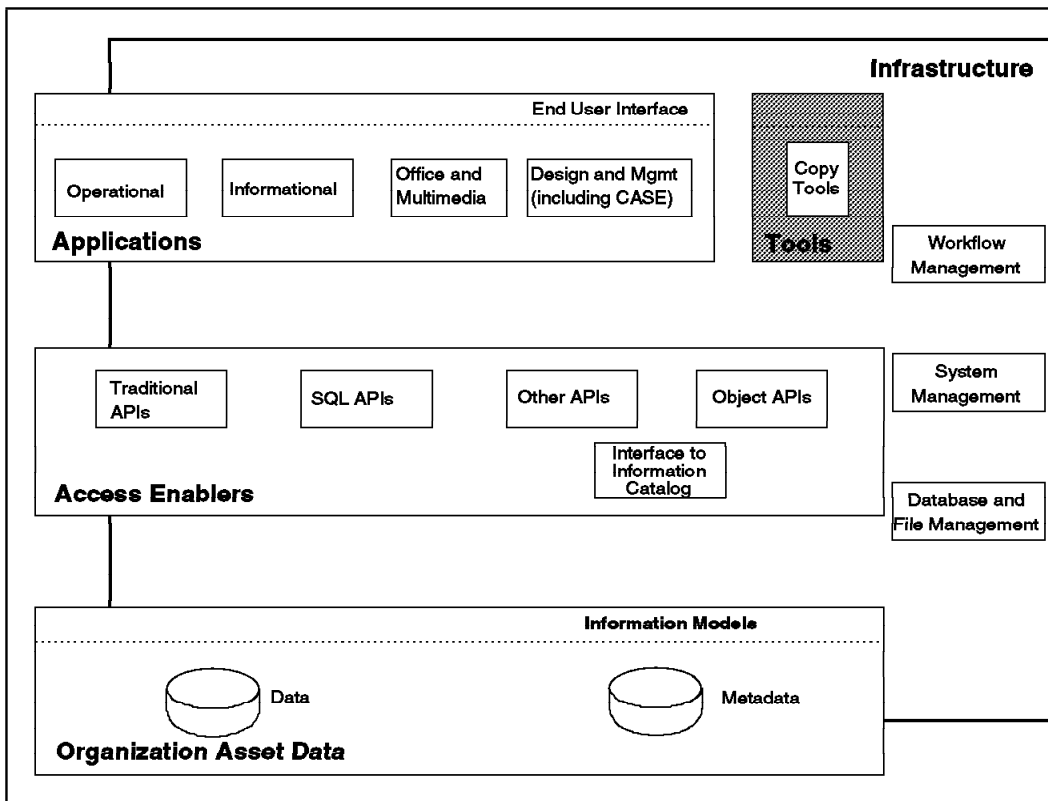


Figure 18. Copy Tools

Data replication tools play in the path from operational to informational data

Data replication tools come in many forms: some are independent products and some are tied to database products. They are heterogeneous with respect to the hardware and software on which they execute; they provide functions such as extract, data conversion, data cleansing, and transformation. They transport data between heterogeneous systems, load data, and apply changed data to target informational data stores. Knowledge workers use informational applications against the informational data thus delivered to apply further analytical manipulation and present the information in graphic form. There is an overlap between informational applications and data replication tools with respect to enhancement of data on its way to becoming an informational object.

Information aggregation level is constantly changing

To get a better feeling for this overlap, we look at information as a spectrum of data aggregation. At the least aggregated end, we start with reconciled data, where there is one row of reconciled data for every record of operational data, for example, an individual bank deposit. The other end of the spectrum is a totally aggregated informational data store, which hypothetically could be a single row representing all deposits for all of the bank's branches. In reality, the spectrum includes aggregation at the branch and region levels by organization and at the daily, weekly, quarterly, and yearly levels by time.

Information is manipulated by knowledge workers and information systems

In practice, knowledge workers need different aggregations of data. An aggregation can be thought of in terms of the SQL statement applied to the source data. The SQL allows column subsetting, row subsetting, and grouping by any column in the table. Aggregations, then, can be any legal SQL statement that conforms to this description. Furthermore, the data can be current or a historical image. A historical image is data generated by transaction activity in a past period of time and kept for the specific purpose of generating a historical aggregation.

Providing information is an endless task for information systems

Some level of aggregation may be common across all knowledge worker groups; this would be a reasonable level of aggregation for the information delivered by the information systems department and the data replication tools. From there on, informational applications or local enhancement programs perform the remaining aggregation. Knowledge worker groups may reasonably go back to the information systems department and ask for further levels of aggregation to be performed by the data replication tools. This leads to a continuous cycle of information systems providing information and the knowledge workers requesting new aggregations. This cycle points out the need for a data replication strategy by which different information aggregations can be generated efficiently.

Finance needs update propagation

Data administrators, work group administrators, and LAN-support staff are the main users of data replication tools. The data replication tool discussion for the finance industry study focuses on the following aspects of data replication:

- Propagation type
- Data delivery technologies
- Data replication tool products.

There are two types of propagation: refresh and update. Refresh propagation examines all data in the base or source and applies corresponding changes to the copy or target, depending on the details of the propagation request.

Update propagation is differential in that data from the base or source is only considered if it has changed since the last propagation. The finance industry has a particular need for update propagation, so this capability of DataPropagator Relational is stressed in this chapter.

8.1 Data Replication Type Requirements

The two basic types of data replication—refresh and update propagation—are data processing solutions to business requirements. This section explores the requirements of the customer information system as a key system needing informational copies of operational data. It then reviews technology terms that underlie the solution provided by DataHub and DataPropagator Relational.

8.1.1 Business Requirement

The customer information system needs its data to be consistent with the operational data within two business days. A customer could apply for one loan at one branch and a second loan at another branch. Even though this may seem unlikely and may be perfectly acceptable, the customer information system must have access to the information of both branches as part of its rating of the customer as a risk and as potential revenue. This requirement can be met through refresh propagation.

Business needs drive the type of data replication

In the refresh approach, a data replication process is executed at least once every two days. The data replication process is made up of at least three specific data processing activities: extract from the operational database, reconciliation and aggregation, and loading into the informational data store. The successful implementation of this business and data processing process is dependent on the process being executed properly at the correct intervals in time. The execution of multiple data processing steps as a single business process is the focus of workflow management. For more information on workflow management, see *Information Warehouse in the Insurance Industry*. However, changing the frequency of this data replication process requires a change in the operational procedures of the information systems department. Refresh propagation requires the full set of data to be processed to reflect the data as it exists at a point-in-time; the data volume can become an inhibitor.

Point-in-time “hardcodes” data replication frequency

Refresh propagation creates a copy of a data object that is consistent as of some point in time. For example, taking a refresh of data for a branch would mean copying data for every account and every customer as of a point in time, say, close of business Friday afternoon. By definition, we know that the data copied represents the precise state of those customers’ transactional information—balance—and nontransactional information—for example, address and phone number—at the time of the copy. This type of copy, then, is very sensitive to data volume; the magnitude of this concern is multiplied by the copy frequency desired to meet the business need.

Data consistency is a business issue

Data propagation minimizes disruption

Operational data—the source of informational data—changes over time. One approach to ensuring consistency requires stopping the transaction systems and then executing the copy process. An alternative is update propagation, wherein the database management system captures data changes—changed data—in a format designed for update propagation. A data replication tool takes the changed data and propagates—applies—the data changes to the copy. This capability, a recent development for relational DBMSs, is less disruptive to the business operation.

Data propagation must be flexible

Update propagation is the preferred approach for frequently updated copies. Update propagation captures updates as they are made to the operational data and subsequently applies them to the informational data object. The propagation software tracks the updates, the capture, and the applying so that updates are not lost. One advantage of this asynchronous approach is that all software components—for controlling, capturing, and applying—do not have to be available at the same time.

Flexibility in frequency is also key

Another important benefit to be gained from data propagation software concerns the frequency of propagation. Data propagation software supports the specification of propagation frequency. That is, to change the frequency of update application from once a day to once an hour, a parameter within the data propagation software is modified. Traditional extracts control propagation frequency outside the software through operational procedures manuals. The same change in the refresh approach would require an operator to run the job more frequently, which implies a change to the operational procedures manual and more room for human error.

Deciding refresh or update is a challenge

The point at which refresh or update propagation is more efficient is not easy to determine. Network transportation and load tool efficiency aside, an update of more than 5%-10% of data generally makes the full refresh more efficient. However, network transportation and efficiency of load tools are critical. Load tool performance in a DB2 environment may vary by a ratio of 4:1, making this criterion an important consideration.

Replication impact must be managed

Copy consistency is one characteristic of copied data. Other generally accepted goals for replicated data include replication transparency and performance transparency. Replication transparency means that knowledge workers are unaware that they are accessing a replica of the data rather than the source itself. Performance transparency means that data access should not be degraded due to the replication or distribution of data.

Consider image, nonrelational, and other types of data

The finance industry has requirements to copy between the following classes of data:

- Structured nonrelational (hierarchical, network, spreadsheet)
- Relational tables
- Architected documents
- Architected images.

Requirements for such copying are identified in the FAA. Copying between these data classes requires transformations, all of which may not be technically possible at this point in time. The Information Warehouse framework products currently support replication of both relational and hierarchical data.

8.1.2 Update Propagation

Users have historically had little choice in refreshing their copied data; the whole object—for example, the DB2 table—had to be copied. As data volumes grow, copying an entire table becomes prohibitively expensive, and clearly undesirable when only a small portion of data was changed within a refresh cycle. In such cases, update propagation is a more efficient approach to maintaining replicated data. Update propagation copying means taking an existing copy and updating the rows that have been recently changed in the source table, to make the copy consistent with the original. DataPropagator NonRelational performs update propagation between IMS/ESA and DB2 for MVS; DataPropagator Relational performs update or refresh propagation between DB2 database management systems.

Data volumes drive the need for update propagation

Update propagation is a more technically demanding approach to data replication. The data changes must be captured from the database management system at which the operational source is being maintained. If update propagation is asynchronous—to minimize impact to the operational applications—then unit of work information must be maintained. This information is used to maintain consistency between the source and target data objects.

Update propagation is a technical challenge

The choice between update and refresh propagation may be based on business or data processing criteria, both of which change over time. Ideally, the choice of propagation method should be dynamic in the software based on system activity. At the least, it should be changeable with minimal intervention by information systems department staff. For more information on data replication, see *Delivering Data to the Information Warehouse*.

Dynamically choose update or refresh propagation

DataPropagator Relational performs refresh propagation of relational tables and is the first Information Warehouse product to support update propagation among relational tables. It propagates DB2 tables in the MVS, OS/400, RISC System/6000, and OS/2 environments and has the unique ability to dynamically choose update or refresh propagation. For more information on DataPropagator Relational, see 8.3.2, “DataPropagator Relational” on page 92.

DataPropagator Relational can decide on propagation type

8.1.3 Copy Consistency

The quality of the decisions being made depends on the quality and consistency of the informational data used to make those decisions. The consistency consideration starts with the operational data, the source for informational data. We assume that the original source operational data is in a consistent state prior to a copy being made. DBMSs make it possible to assume a state of consistency at a given “quiesce” point. The quiesce point, or stopping of database activity, ensures that there are no transactions currently changing data in such a way that the data being copied is inconsistent. For example, a customer could be transferring funds from a savings account to a checking account. The quiesce point ensures that the data reflects the state of the accounts either before the transfer is made or after, but not in the middle of the transfer.

Consistent data reflects the state of the business

Update propagation changes business possibilities

Copies have traditionally been run at quiesce points, which typically correspond to a business event such as the end of a business day. Informational analysis, then, is performed against informational data that lagged behind the corresponding operational data. Most of the time, this is an accepted, even desired situation. For example, certain government filings require numbers representing the financial standing as of the end of the financial year. Update propagation, however, opens up new possibilities for copy consistency with respect to the operational data.

8.1.4 Update Sequence

Updates to operational data are maintained in time

The quality of data copies—the informational data—depends on the integrity of the data. Update propagation is possible because the DBMS maintains a log wherein all database updates are serialized and maintained in time sequence. The serialization is assisted by database locking. An updating transaction acquires an exclusive lock on a piece of data as a prerequisite to updating the data. The lock taken persists until the transaction either commits or rolls back.

Serialization of access means update in commit order

The lock ensures that another transaction will not change this piece of data until the lock-holder transaction terminates. This serialization of access to a given database row does not necessarily guarantee correspondence to the order of transaction commit time. It is quite possible for one transaction to start later and commit earlier, even if both update the same database row, as shown in Figure 19. This independence of row update, commit point, and propagation of corresponding log records presents a challenge to maintaining copy consistency.

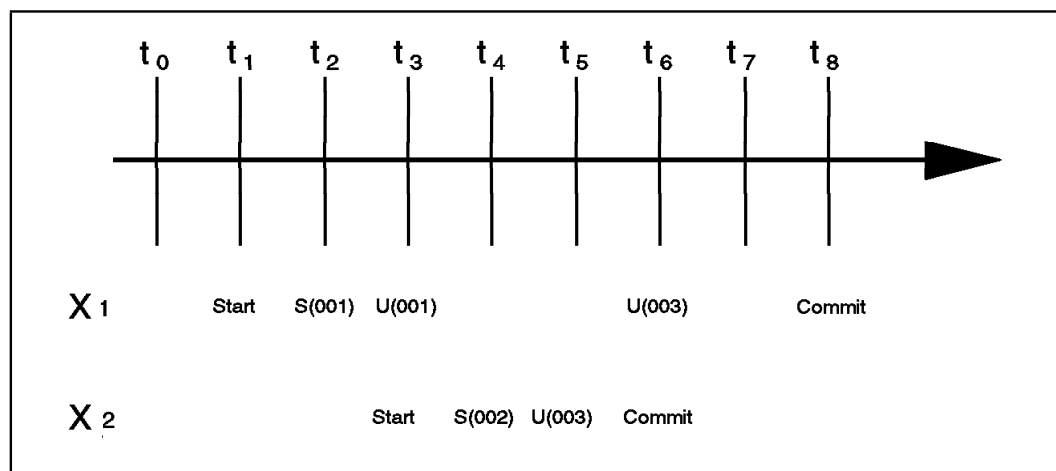


Figure 19. Transaction Time Line

The log records for the transaction database activities shown are transmitted to a copy site as they occur. This scenario shows an inconsistency in the copy for at least the time period from t_6 to t_8 . The copy site has the committed update of X2 and believes that to be the accurate state of row 003. The copy site has an uncommitted update of X1. It is clear that at t_9 , with all log records propagated, the value for row 003 will be that after the X1 update. However, at t_7 the copy clearly has the wrong value. This is a direct result of the log propagation approach. Furthermore, any system failure can delay the

commit for X1 indefinitely, leaving the copy inconsistency for a significant time period.

The *order* of update in the database must be the same as the *order* of transaction commit. This correlation forms the basis of our understanding of update semantics. For example, assume a transaction makes a buy or sell recommendation for a particular stock or for a group of stocks. It stores the recommendation for each stock in a row of table RCMDTN. It builds the recommendation list for all stocks first and then proceeds to update the RCMDTN table, one row at a time. The transaction was executed in batch mode. The recommendation for stock XYZ was to buy, based on the available information.

Updates and commit sequence affects data quality

Because of heavy trading activity in this stock, the transaction was rerun again for this stock only. This transaction changed a recommendation to “sell” and finished before the batched one. It acquired and released a lock before the batched transaction processed the particular stock. The “sell” status was quickly overwritten (and lost) by the batched transaction, which terminated later. However, when propagating this information, “sell” status persisted for a while, because of the timing of the transaction’s execution. The sum result of these events was incorrect informational data: our transaction was not designed properly, because it in effect allowed—in a semantic sense—for a lost update. The existence of a “lost” update was short-lived in the original; in a copy it became more pronounced because of an intervening snapshot timeframe.

Copy propagation can lose an update

Let’s assume that a transaction, T, selects 10 “best buy” stocks. It executes twice, referred to as transaction occurrence Tx1 and Tx2, in that order in time. Because of markedly changing conditions in a short period of time, the two occurrences generate different (nonoverlapping) lists of stocks. Tx1 commits before Tx2, in the order in which they were submitted. Figure 20 on page 90 shows this sequence of transaction executions.

Update propagation changes update semantics

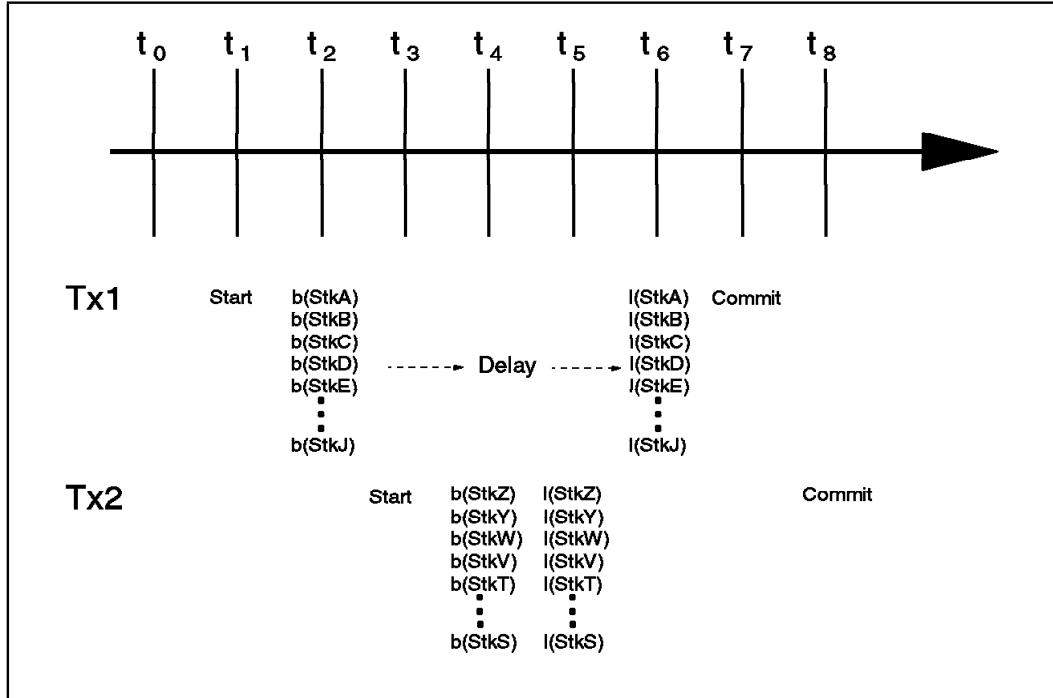


Figure 20. Stock Pick Transaction Sequence

Update propagation changes update semantics

Now, let's assume that Tx2 actually updates the database before Tx1 as Tx1 is momentarily delayed before commit. Update propagation intercepts updates in the order of updates being actually written, NOT in the order of transaction commit. Therefore, in the copy process, stock recommendation Tx2 will come up before Tx1. In terms of trend analysis, this is quite incorrect and inconsistent with the original database. This scenario highlights the importance of time sequence in database updates and update propagation.

8.2 Data Replication Technologies

Data delivery technology becomes IW data replication

Data delivery products predate the Information Warehouse architecture and will continue to evolve with that architecture. The architecture will make it easier to manage the delivery process and reduce the cost of both tools development and execution. We have focused on a particular data replication environment and set of products in this solution thread. This set of products is part of data delivery technologies that include the following:

- Data access protocol
- Update propagation
- Refresh propagation
- Archive.

These technologies are discussed in the sections that follow.

8.2.1 Data Access Protocol

Data access to remote systems is enabled as a part of database function. IBM has published DRDA as an open interface for remote access to relational databases. The architecture supports both remote unit of work (RUW) and distributed unit of work (DUW), the latter being supported between DB2 Version 3 systems only. Other vendors are implementing RUW support in their DBMSs.

Our solution thread takes advantage of the DRDA protocol through the use of DDCS/2. DataPropagator Relational apply programs request changed data from a remote location. The DataPropagator Relational apply programs are acting as an application requester, using DDCS/2 connection to the database containing the data to be propagated.

8.2.2 Update Propagation

Update propagation depends on a DBMS log interface to capture database changes. The database management system writes records to the log as a normal recovery function. The record format is slightly different for tables identified for propagation. The DataPropagator Relational capture program reads these log records and stores them in a table. This scheme—reading changed data from the log—minimizes the performance impact on the operational application. The DataPropagator NonRelational product between IMS/ESA and DB2 for MVS uses update propagation, driven by logged changes, whether propagating from IMS to DB2 or from DB2 to IMS.

*Minimize
impact to the
operational
application*

8.2.3 Refresh Propagation

A variety of tools exists that implement refresh propagation. DataRefresher performs refresh propagation on a variety of sources, including IMS, VSAM, and DB2 for MVS. Refresh tools are often extended by specialized code available as exits. DFSORT is a tool that illustrates this feature. It is fast and economical and performs sorting and manipulation of data. The Information Warehouse framework looks to the exits to implement reconciliation and enhancement as required by the business needs.

8.2.4 Archive

Archive systems are a distinct class of data copying and data delivery, as they specifically manage historical data. DB2 Data Archive Retrieval Manager/MVS* (DARM) supports this type of data delivery. Archive products will become more important as historical data and the trend analysis against it become more popular.

8.3 Data Replication Products

Two IBM software products—DataPropagator Relational and DataPropagator NonRelational—fulfill functional requirements of the Information Warehouse architecture. DataPropagator Relational is a product that fits on top of DataHub and manages refresh and update propagation. DataPropagator NonRelational is a product that manages propagation of IMS data. The DataHub product provides functional support for system management in the Information Warehouse architecture. A brief discussion of DataHub is followed by a more lengthy discussion of DataPropagator Relational.

8.3.1 DataHub

DataHub is an integrated database management system facility that enables a database administrator (DBA) to manage IBM's relational database management system family (DB2 for MVS, SQL/DS for VM, DB2/400, DB2/6000, and DB2/2). These relational DBMSs are managed from DataHub on an OS/2 workstation, providing a central point of control for all participating hardware and software platforms where data resides.

DataHub greatly simplifies administration tasks. With it, the DBA can:

- Display database objects and their relationship to other objects
- Copy database objects between or within DBMSs
- Invoke relational DBMS utilities
- Manage authorizations that permit user access to databases
- Display status of relational units of work across platforms.

DataHub provides a single, consistent, seamless interface to copy tools—both IBM and vendor products—from the programmable-workstation point of control. This is achieved by using Information Warehouse architecture-defined and DataHub-supported interfaces. Specific tools such as DataPropagator Relational are needed to perform copy tasks. DataHub/2 provides a unified launching and control platform. This solution platform responds to an industry need for a flexible management facility with a single point of control.

8.3.2 DataPropagator Relational

Customer surveys and academic research identified functional requirements for data replication and, in turn, DataPropagator Relational. DataPropagator Relational is an Information Warehouse copy tool that runs as an action in the DataHub object-action paradigm. Its primary function is to propagate operational data to informational data stores for knowledge workers using informational applications to access the informational data. The Information Warehouse architecture defines two types of propagation: refresh and update. Table 4 on page 93 shows how refresh and update propagation is supported by the DataPropagator Relational products for the respective DB2 family members.

Table 4. DataPropagator Relational Propagation Paths					
		Copy Server (Target)			
		MVS	OS/400	RISC Syst./6000	OS/2
Data Server Platform (Source)	MVS	U, R	U, R	U, R	U, R
	OS/400	U, R	U, R	U, R	U, R
	RISC Syst./6000			R	
	OS/2				R
Legend:					
R Refresh propagation					
U Update propagation					

DataPropagator Relational has facilities to define, synchronize, automate, and manage copy operations from a single control point for data access across the enterprise. This control point is a DataHub/2 workstation. DataPropagator Relational can be used to tailor or enhance data as it is copied, resulting in detailed, subset, summarized, or derived data on the desktop. DataPropagator Relational provides these capabilities through three independent components, as follows:

- Capture** The capture component reads log records from the database management system in which the source table is defined.
- Apply** The apply component reads data from either the changed data table or the source table and applies that data to the copy table. Data is read from the changed data table for update propagation and from the source table for refresh propagation.
- Administration** The administration component is used to define the tables involved in propagation and the propagation processes.

Figure 21 on page 94 shows the components of DataPropagator Relational.

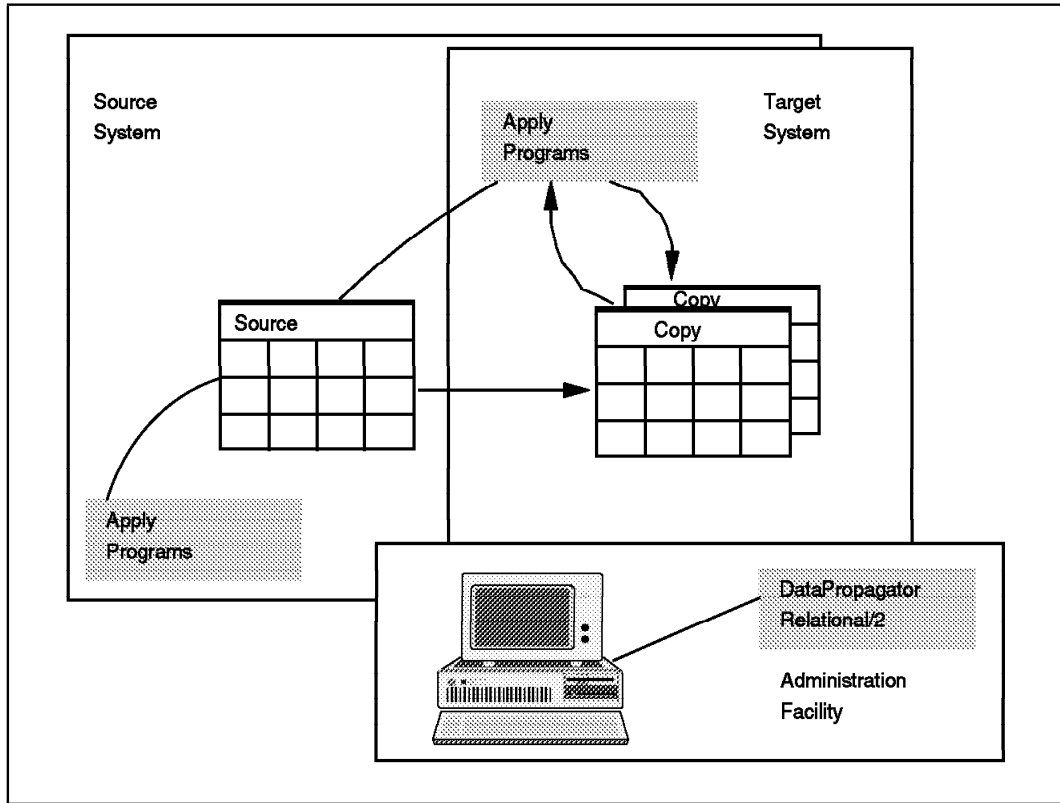


Figure 21. DataPropagator Relational Components

The DataPropagator Relational family of products implements these components on a variety of platforms. Table 5 shows the DataPropagator Relational components and their product implementations.

Table 5. DataPropagator Relational Components and Products	
Component	Products
Capture	<ul style="list-style-type: none"> • Capture/MVS • Capture/400
Apply	<ul style="list-style-type: none"> • Apply/MVS • Apply/400 • Apply/6000 • Apply/2
Administration	DataPropagator Relational/2

8.3.2.1 Server Structure

DataPropagator Relational defines three *logical* servers. These servers should not be confused with the three components—Capture, Apply, and Administration. Figure 22 on page 96 shows the three *logical* servers, as follows:

- Control Server** The control server is the location where copy registration and subscriptions occur. This server also provides a focal point for launching copy requests. DataPropagator Relational/2 is the definition-time component of DataPropagator Relational, responsible for control functions.
- Copy Server** Copy Server is the system where the target copy is to be created and maintained. The DataPropagator Relational Apply maintains the target copies; it usually runs on the copy server. The DataPropagator Relational Apply may run on the data server and use DRDA definitions to add data to the target copy.
- Data Server** The data server is the system holding the original, base table. The capture program is the runtime change capture component of DataPropagator Relational and resides at the data server.

The DataPropagator Relational software components—capture, apply, and administration – can thus be configured in several ways with respect to the servers.

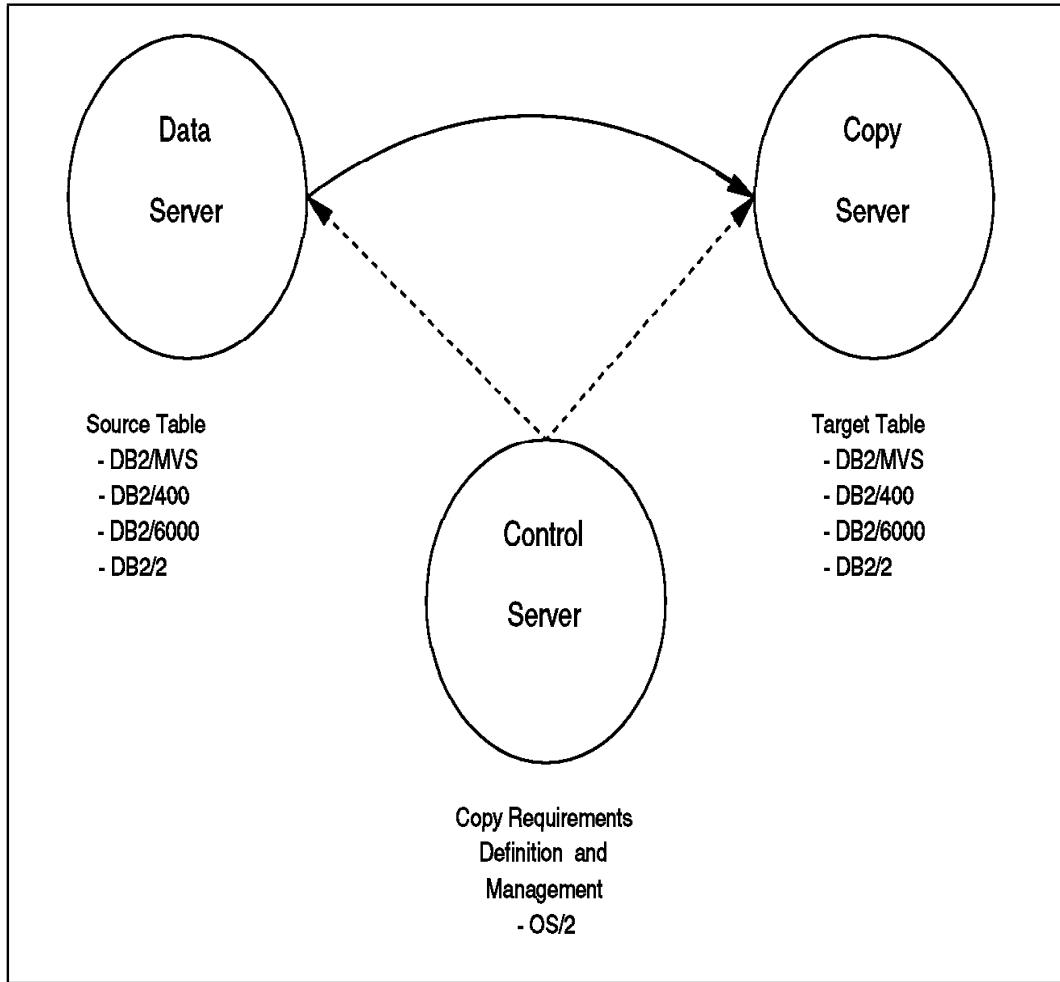


Figure 22. DataPropagator Relational Logical Servers

*The DPropR
is flexible*

DataPropagator Relational's server structure allows for a great deal of flexibility. The central piece of the architecture is DataPropagator Relational/2 running at the control server. It is responsible for the definitions of data sources and copy requests and is also the launch point for the copy activities. DataPropagator Relational/2 implements one of the interfaces for data replication identified in *Information Warehouse Architecture I*: the interface to the Object Handler Meta-data. The Object Handler, as defined in the Information Warehouse architecture, provides a user interface that acts as the end user's point of interaction with all data replication tools. It presents a listing of common and tool-specific meta-data, viewed as objects, and a mechanism for selecting and acting upon the meta-data.

Control Server

The control server makes administration and navigation tasks easy for the data replication administrator. It also eases the complexity created by having data replication tools from different vendors.

Copy Server

The copy server holds the target table. Its function is to transport refresh or changed data from the data server to the copy table. The copy server task is implemented as the DataPropagator Relational Apply and is simplified by the changed data, compliant with the Data Staging Interface as defined in *Information Warehouse Architecture I*, that is the input to Apply.

Data Server

The primary functions of the data servers are capturing changed data for update propagation and providing source table data for refresh propagation. The implementation for the changed data capture function is tied to the technical specifications of the DBMS in which the original base data resides (for example, DB2 for MVS). The changed data capture function must be able to operate on log records being written to the DBMS's log. In the case of DB2 for MVS, the changed data capture implementation—Capture program—uses the DB2 Instrumentation Facility Interface to capture log records from the DB2 log.

The Data Staging Interface defined in *Information Warehouse Architecture I* defines a common form for making changed data and full refreshes available to tools that maintain reconciled or derived data. It makes the task of data replication tools easier by establishing a common format for representing changed data; the tools do not have to concern themselves with a multitude of data formats as input or output data objects. DataPropagator Relational maintains a set of relational tables capturing data changes for registered base tables.

8.3.2.2 DataPropagator Relational Capture Program

Intercepting changed data requires a run-time component tied to the DBMS. DataPropagator Relational supports DB2 for MVS, DB2/400, DB2/6000, and DB2/2 as data servers, though only DB2 for MVS and DB2/400 data servers can support update propagation. The Capture program uses specific DB2 logging and monitoring facilities to capture changed data.

Functional Description

DB2 for MVS records every SQL transaction in a log for diagnostic and recovery purposes. These log records are available to application programs through the Instrumentation Facility Interface READS request. Recovery records for tables with the Data Capture Changes attribute contain a full before and a partial after image of the table row in the log record. The Capture program reads the DB2 active log to detect any changes in a specified user table and captures the changes in a staging table known as the changed data table. These captured changes are pulled by DataPropagator Relational Apply and applied to its copy of the user table (a snapshot) to maintain data currency.

The Capture program can capture changes for more than one user table. For each source site, the names of each user table and its associated changed data table are specified in the changed data control table. In addition, the Capture program requires the pruning control table and the critical section table for communication between itself and DataPropagator Relational Apply. The user must define a tuning parameter table in which performance tuning information is specified.

The Capture program starts the monitor trace using the DB2 Call Attach Facility and reads the log to detect changes to the user tables. As the user tables are updated, the updates are written to the DB2 active log. The Capture program monitors the updates to the user table on the DB2 active log. It then collects the user table changes in the staging table. Updates are inserted in the staging table for the DataPropagator Relational Apply's use.

Pruning of the staging tables means discarding data that has already been applied at the copy server. This activity is controlled by updates made by DataPropagator Relational Apply. The staging tables are pruned by the Capture program as data is copied by the DataPropagator Relational Apply. The pruning control table is used to communicate pruning control information. Capture performance tuning may be controlled by the user by means of the tuning parameter table. Figure 23 on page 99 shows the data flow of the changed data capture process at the DB2 data server.

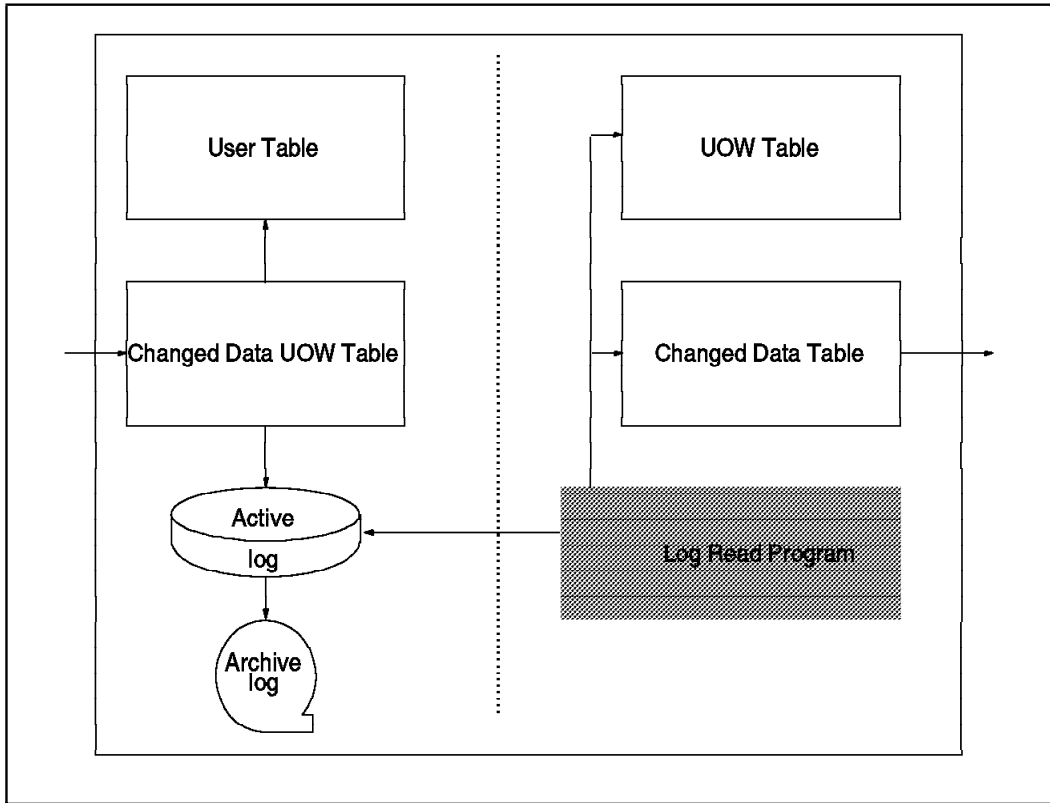


Figure 23. Changed Data Capture

Active log configuration

DB2 has a two-tiered structure for its logs: active logs and archive logs. The active logs are defined as a pool. When an active log is filled, DB2 switches to the next active log and schedules the log just filled for archive processing. This log will not be reused until the archive process is completed. In time, after completing a round-robin cycle of active log data set usage, DB2 comes back to the now archived active log and overwrites its content. The DB2 log structure is shown in Figure 23.

Capture program is compatible with DB2's log structure

At present, the Capture program can only read from active logs. Consequently, changed data is available to the Capture program in a window of time between when the log record is written and the active log data set is reused. The duration of that window depends on the active log configuration, the use of the DB2 ARCHIVE command (forced log switch), and update activity in the system.

8.3.2.3 DataPropagator Relational Apply

DataPropagator Relational is flexible on its sources and targets

DataPropagator Relational Apply copies source tables from the source site—data server—to the target tables. In addition, this program can apply column functions—for example, SUM and AVG—to the source table; the result is appended to the target table. The target system can be a DB2 database as shown in Table 4 on page 93; the same DB2 instance can be both source and target. There can be any number of DataPropagator Relational Apply instances running on a copy server, and any number of copy servers in the DataPropagator Relational data distribution network. Several tables control the operation of DataPropagator Relational Apply. The administrator creates these tables and specifies their content through DataPropagator Relational/2.

Control Tables

Tables control server communication

The DataPropagator Relational Apply executing at the copy server needs to know which servers control its operation. The linkage is provided via a routing table. Snapshot definition is located in a refresh control table located at the control server. In addition to snapshot definitions, it has a global control record, which allows for the control of the associated refresh control table. This allows snapshot definitions to be disabled and thus ignored by the DataPropagator Relational Apply. The DataPropagator Relational Apply can be turned off by a switch in the global control record.

DataPropagator Relational Apply is directed by a snapshot definition. This specifies among others, the:

- Base table, its name, and location
- Copy table, its attributes, and structure
- Refresh policy
- Enable/disable flag.

A base table to the snapshot must exist and its structure and attributes must be registered through DataPropagator Relational/2. The control tables and copy table(s) must also be created and set up through the DataPropagator Relational/2 before the DataPropagator Relational Apply can be started.

Table Copy Control Attributes: Each source or copy table has a set of attributes defined. They control the content and consistency characteristics. The attributes are specified in the Changed data Control table and Refresh Control table, and have the following meanings:

Condensed The condensed attribute controls what is stored in a table. The condensed table does not have historical data.

Complete The Complete attribute dictates whether full refresh can be performed. If the source table has been defined with the Complete attribute, full refresh can be performed.

Consistent The Consistent attribute controls the consistency level. Consistency can be at either the convergent or transaction level. The convergent level contains an unbroken sequence of committed, uncommitted, aborted, and compensating updates. The transaction level contains an unbroken sequence of committed updates only. User tables are always assumed to be transaction consistent, as are consistent changed data tables.

Base and trend aggregate tables are restricted to being transaction consistent.

DataPropagator Relational Apply takes advantage of the source-to-target refresh rules that can be derived from the structures and attributes of the source and target tables. In particular, the DataPropagator Relational Apply automatically chooses a refresh algorithm (refresh or update propagation) as well as the optimal source (user table, changed data table, consistent change data table).

Attributes influence copy algorithm

Consistency

In 8.1.3, “Copy Consistency” on page 87, we discuss briefly the issues arising from intercepting data changes through the log interface. We identify two types of exposure: one exposure is associated with uncommitted and aborted changes, and the other is a semantic inconsistency, caused by a timing difference between an update and the corresponding transaction commit. One way of ensuring data consistency is by grouping changed data by its unit of work (UOW) identifier. The UOW identifier is only written when the transaction commits. An equijoin can be performed against the changed data and UOW tables (and their rows) to filter out aborted, uncommitted changes. In addition, ordering by UOW time stamp restores semantic consistency.

Consistency is managed through tables and SQL

Data Staging

Often there is a need to derive multiple copies from the same original source. In our solution, all branches have a subset of the master customer information copied to them. Yet, from the performance, consistency, and operational points of view, it is better to intercept changed data once than to have multiple processes doing the intercepting. The data staging approach allows the changed data to be intercepted once and multiple target tables to be populated from it.

Staging: fan-out copies, consistency

8.3.2.4 DataPropagator Relational/2

Data replication administrator interaction with DataPropagator Relational occurs through DataPropagator Relational/2. DataPropagator Relational/2 uses the definitions generated by this interaction to create and maintain DataPropagator Relational control tables for the three DataPropagator Relational components. DataPropagator Relational/2 governs three types of administrative activities, as follows:

DataPropagator Relational/2 is the entry point to copy control functions

- Authorization
- Registration
- Subscription.

These activities are part of the administrative task for data replication.

Authorization

Authorization controls administrator access to control tables

At some point DataPropagator Relational Applies would need to access internal DataPropagator Relational tables created during registration. Authorization enables user IDs associated with DataPropagator Relational Applies to access these tables. Assistance is provided as an “action” from DataHub/2 to automate the authorization process.

Registration

Registration controls the candidacy of source tables

Registration identifies the tables that can be used as a source table for copying. The two object types defined are current and candidate registration. Current registration tables can be used as sources for copying. Candidate registration tables are visible to the DataPropagator Relational administrator or registrar via DataHub and can be changed to current registration.

Subscription

Subscription controls the copy activity

Subscriptions define the propagation activity. The two object types defined for subscriptions are current and candidate subscriptions. Current subscriptions control the copy activity to be done and when it is to be done. Candidate subscriptions are visible to the DataPropagator Relational administrator, registrar, or subscriber via DataHub and represent registration entries that have not been used as source tables for copying.

8.3.2.5 Security and Audit

DataPropagator Relational leverages existing security and audit function

DataPropagator Relational makes use of the security and auditability features of the MVS, AS/400, and OS/2 operating systems and the DB2 DBMSs. Integrity of host, database, and workstation security procedures is preserved. DataPropagator Relational works in conjunction with the security software, including the use of user IDs and passwords in OS/2 and large servers and databases. In addition, DataPropagator Relational makes use of existing communications function between DataHub and remote hosts and between source and target databases. All authorizations at the database are controlled by the database manager based on the authorizations granted to those user IDs.

8.3.2.6 Pruning

Pruning manages changed data growth

The changed data table has a potential for unbounded growth; pruning of old or unnecessary data is necessary. The Capture program does the pruning, because it maintains the changed data. The pruning is based on the information in the pruning control table. Consistent change data is controlled outside the Capture program.

8.3.2.7 Tuning and Control

Automation of data replication is DataPropagator Relational's ultimate objective. However, experience with implementing automation has shown that users need to override some aspects of automation. They also need to have a good understanding of how the process operates. Among the controls and tuning options DataPropagator Relational provides are the following:

Automation is the ultimate goal

- Statistics (rows fetched, inserted, and the like)
- Differential or full refresh choice (update versus refresh propagation)
- Full refresh enable/disable switch for each source table
- Absolute retention duration, which overrides the automatic changed data pruning mechanism
- Flexible commit frequency of LRP.

8.3.2.8 External Sources

Corporate data residing in data stores other than directly serviced by DataPropagator Relational can be interfaced to DataPropagator Relational and thus propagated using a store-and-forward approach. This applies in particular to IMS and VSAM data stores. DataRefresher and DataPropagator NonRelational integrate with DataPropagator Relational through the data staging interface to support copying IMS and VSAM data to DB2 for MVS, DB2/400, DB2/6000, and DB2/2 copy tables.

DataPropagator Relational can accommodate other source data types

8.4 Implementing the Solution Thread

Implementing the solution thread requires the following steps:

1. Define customer information and profit analysis tables in a DB2 environment on the host.
2. Resolve naming standards and authorization IDs needed for distributed communication.
3. Enable a process of populating these tables.
It is quite likely that tables will be populated from multiple data stores, most likely nonrelational. This makes update propagation of base customer information impossible. The base data will have to have a full refresh.
4. Define a matching set of tables in a DB2/2 environment.
5. Use DataPropagator Relational/2 to register base tables in DB2.
6. Use DataPropagator Relational/2 to subscribe copy subsets and authorize programs.
7. Set request execution frequency and pruning triggers.

Chapter 9. Conclusions

The finance industry has been deeply affected by global business conditions and innovative use of information technology. The outlook for the future suggests more competitive pressures and exposure to global economic conditions. The industry response is to formulate strategies which emphasize the following:

- Customer relationship
- Understanding and leveraging risks
- Moving to fee-based revenue
- Containing costs
- Redefining branch function.

Strategies address a multitude of corporate needs

Some of these strategies, presented at a high level of abstraction, are neither new nor unique. What makes them new is a compelling need to manage strategies at a different level of detail: a financial institution needs to track its key business indicators on a weekly or even daily rather than quarterly basis. It needs to understand its customer set at the individual level, and it must be able to use this knowledge when transacting with its customers. Exposure to a global economy means that interest rates worldwide must be monitored closely, and implications of their alignment understood. The Information Warehouse architecture facilitates such management. It defines, in a generalized way, applications, access enablers, organization asset data, tools, and infrastructure to help solve the information delivery problem. It defines interfaces and data representation formats for integration and openness.

A new level of detail is required

Business innovation calls for a realignment between business and information technology providers. Whereas in the past, high-level planning was deemed enough, today the demand for information necessitates contacts at all levels between business and information technology staff.

Business and information technology must integrate better

Key information systems emerge as critical to meeting finance business objectives in the 1990s. They are:

- Customer information
- Risk analysis support
- Profitability analysis
- Asset and liability information.

Building these systems challenges the present technology. These systems need an enterprise view of data, requiring the integration and reconciliation of operational data from multiple lines of business. The key systems are complex, involve large data volumes, and incur high development costs.

Deployment is a major effort, possibly requiring a new technology at the branches such as client-server, and may take years to accomplish.

A structured approach is necessary

It is doubtful whether these systems can be built, deployed, and maintained at a reasonable cost, without a generalized, architected way of gaining access to information. The generalized approach calls for use of architectures, models, standards, and interfaces at both the enterprise and industry levels. The Information Warehouse architecture offers a structured approach to informational data access, and FAA offers it at the industry level.

At the enterprise level, this structured approach calls for understanding and organizing the organization asset data; building a master plan for data and information. It also implies a careful assessment of processes and source data needed to support the systems deployed for information delivery.

FAA is the industry guide to strategic solutions

The Financial Services Data Model offers the model of the data and business activities at the industry level. It includes roughly 80% of the average finance enterprise's data and processes. It provides a mapping into the enterprise's information technology and attempts to shield business logic from changes in rapidly changing technology. IBM has developed the Financial Services Data Model according to FAA standards. Use of the basic model with customization can save individual organizations time and money and help achieve data and systems integration. The questions of how far to proceed with the model implementation and when to control the actual data representation (for example, database formats and program copybooks), remain open and need to be decided by individual organizations.

IW architecture is the generalized approach for information delivery

The Information Warehouse architecture satisfies the industry requirement for a generalized, structured approach to data access. Information is context dependent; it is more a process of being informed, than passive data storage or presentation. Key in the process is the end-user environment where data can be massaged and correlated. Today, workstation and work group environments are best suited for information acquisition. The Information Warehouse architecture defines standards and interfaces in this environment.

Openness

The finance industry needs open standards to facilitate interoperability with respect to diverse platforms, while keeping location and data representation transparent to the knowledge worker. The Information Warehouse architecture provides the basis for bringing together products and tools from many vendors. For the finance enterprise, it reduces the cost of gaining information, accommodates diverse requirements within one generic solution, and simplifies systems management.

Data replication helps meet needs for diverse information

Meeting diverse information needs requires an effective data replication strategy. An effective data replication strategy has provisions for automated copying, a single workstation point-of-control, and both full and differential refresh. New Information Warehouse products that contribute to those strategy provisions include DataPropagator Relational, DataHub, and DataRefresher.

Information is context and people dependent. The most productive environment for information acquisition, for knowledge, is the programmable workstation running decision support software, with transparent access to a wide variety of data stores and navigation assistance through that maze of data stores. This is the direction of the Information Warehouse framework and its set of products.

*Desktop
access*

Appendix A. Models and Modeling

The solution presented in this book devotes significant attention to models and modeling. Modeling has been given much publicity over time and has been considered crucial to a well-organized and efficient data processing organization. However, modeling has, in general, failed to deliver on the promised benefits of model-based application generation. Some of this failure is due to the sheer variety of modeling methodologies available, and some is due to the esoteric language and complexity of the concepts inherent in modeling. Perhaps the largest contribution to this failure is the lack of open interfaces and automation between the components and phases of the application development life-cycle. In this appendix, we present the concepts and benefits of modeling by a simple example and lay the groundwork for the role of the Financial Application Architecture* (FAA), Insurance Application Architecture* (IAA), and Retail Application Architecture* (RAA), in data processing in general and Information Warehouse implementation in specific.

Modeling organizes the data processing environment

A.1 The Construction Model

The building of a structure—a home, an office building, or other complex structure—serves as a platform for discussing the benefits of a model and modeling. A model is an abstract representation of a real world environment. Modeling classifies certain aspects of building into things called entities. The concept of entities immediately presents a challenge for relating modeling to a real-world environment. Entity is a term used to classify people, places, things, ideas, concepts, or events that are relevant to the business. The key here is to understand the motivation for entities. Entities provide a way to group things that have common characteristics or role with respect to the business.

Entity: classifying things

For example, within a construction company, entities include nails, boards, and windows; carpenters, plumbers, and electricians; contracts; trucks; and other components of the business. As a general categorization, entities is a convenient catch-all for anything with which the construction project leader—the general contractor (GC)—has to be concerned. The benefit is that the GC can look at one list of things needed to build the house.

Entities can be grouped to simplify their use

Further reflection reveals that these entities are not all the same, that some subsets of these entities have common characteristics. If it is of benefit to group all entities, then it is of more benefit to group them so that the common aspects of the subgroupings can be utilized.

A.1.1 Entity: Things

Entities make the overall process simpler

The next step is to create entities within the larger-scope Entity, based on these common aspects or ways of being used. For example, nails and boards have attributes in common: they are both things to be ordered, stored, and physically incorporated into the structure. We therefore create a specific type of entity called Materials. Materials is a grouping of things that are handled in a similar manner from a business perspective and typically have common attributes. The attributes describe the nature of the entity. In this case, the attributes are the color, size, and other physical aspects of the thing. The value here is that the GC can use one order sheet for all things that fit into the Entity category Materials. The GC can use another form for all subcontracts for services. The GC's job is now easier because the different parts of the job can be generalized.

Entities help generalize data processing processes

At this point, we have introduced two perspectives on things essential to the business: the way these things fit into the business—what the business does with the thing—and the attributes or descriptions of these things. The benefit then is that we can think of the things that are part of our business as a general group. We can also generalize the business activities performed against the things.

A.1.2 Entity: Agreements

The next set of entities is centered around the contract, or agreement, legal or otherwise, that is part of building the structure. Contracts are written, agreed to, and enforced. Contracts are different from nails and boards, which are ordered, installed, or are part of the physical structure. By separating these two sets, we can treat them appropriately from a business point of view. What is of more interest here is that they can be treated consistently from a data processing point of view. That is, application code can be written to consistently operate on data about things defined as being the same entity. Furthermore, application code written to operate on data about things used in building a house is consistent with application code written to operate on data about those same things used in building an office.

This approach suggests that contracts in the construction business, categorized as an entity called agreements, can be viewed from both a business and a data processing perspective as similar to contracts in the insurance industry, where they are policies.

A.2 The Annual Report As a Model

We can then extend these concepts to the corporate statement. The annual report presents the financial status of an enterprise in terms of its assets and liabilities. Both the assets and liabilities can be seen as entities, but this is still rather ambiguous with respect to common experience. A better example is the subheading Plant and Property under assets. This is a financial view of things the enterprise has as an asset. This perspective on the enterprise is similar to the points we stressed as being a benefit for modeling, entities, and the general categorization philosophy of modeling. Regardless of the industry within which the enterprise does business, the enterprise always has some type of plant and property.

The annual report is a model

From an industry perspective, we can treat all plant and property as something that has value, that exists. From a data processing perspective, we can expect to write applications that sum up the present value of that plant and property. Furthermore, we can expect to write applications that depreciate that plant and property over time. The treatment and expectations of these business objects are independent of the enterprise's industry. We have gained perspective on a component of the business and have gained an opportunity to leverage data processing resources by this generalization, which is wholly compatible with the objectives of modeling.

A.3 Information Warehouse and Modeling

Models contain representations of the enterprise's business in the form of Entities and other modeling constructs. These constructs contain technical and descriptive information about the business objects. The technical information includes data type and length and may in fact be used in limited ways by an application generator. The descriptive information is for reading and understanding purposes only for the model user. This descriptive information is called meta-data and is a crucial part of an the Information Warehouse environment.

Models are a formal representation

Meta-data explains the meaning of the object and the data processing object in business terms. It helps the administrator know whether the business/data processing object is needed for informational analysis. It also helps the knowledge worker understand the meaning of the object, once it is incorporated into the Information Warehouse implementation. This connection between modeling, the model, and the dictionary for the knowledge worker (the Information Catalog) is dependent on a subset of the model information in the Information Catalog.

Meta-data explains object meaning

The descriptive information also reflects the reconciliation and enhancement process. It describes the business/data processing object as it exists. The knowledge administrator and the business analyst know what the informational data needs to look like for use in an informational environment. The processes of decoding, reconciling, and enhancing operational data for use in an informational environment are based on these before and after descriptions.

Meta-data reflects data enhancement

List of Abbreviations

ATM	automated teller machine	GOI	generic output interface
CPU	central processing unit	GSA	General Store Application
DASD	direct access storage device	HHT	hand-held terminal
DBA	database administrator	IAA	Insurance Application Architecture
DIS	data interpretation system	IBM	International Business Machines Corporation
DRDA	Distributed Relational Database Architecture	ITSO	International Technical Support Organization
EDI	electronic data interchange	MIS	management information system
EFT	electronic funds transfer	PWS	programmable workstation
FAA	Financial Application Architecture	ROA	return on assets
GMROI	gross margin return on investments	RAA	Retail Application Architecture
		UPC	universal product code

Index

A

- access enablers
 - FAA 40
 - Information Warehouse architecture 56
 - SQL mappers 56
- aggregation 84
- airline industry 18
- application
 - data access language 56
 - generator 26
 - informational 4, 56
 - isolation 37, 40
 - layer 37
 - narrow focus 23
 - nonarchitected 51
 - operational 21
 - reengineering 25
 - series 23
 - shrink-wrap 58
 - submodel 42
 - update propagation 74
 - user interface environments 37
- application development 25
- architecture
 - benefit 37
 - industry 5
 - Information Warehouse 5, 30
 - longevity 23
 - scope 33
- asset and liability analysis 21
- authorization 102
- automated teller machines 17
- automation
 - data replication 61, 83

B

- balance sheet management 13
- branch office 15

business

- integration 19
- networking 17
- business success factors 13

C

- Capture program 98
- changed data 77
- channels 14
- client warehouse
- client-server
 - consortium 67
 - leverage 67
- commit serialization 88
- commodity market 16
- computer-aided software engineering 26
- consolidation
 - back office 14
 - enterprise 16
- control server 97
- convergent consistency 100
- copy server 97
- copy tools 63
- core logistic 18
- cost control 14
- credit rating 71
- customer contact 68
- customer information systems
 - data source 70
 - requirements 74
 - use by analysts 20
 - value 66
- customer relationships 13
- customer service 14

D

- data
 - consistency 87

- data (*continued*)
 - historical 28, 84
 - meta-data 27, 77
 - role 12
 - what exists
 - Information Warehouse architecture goal 60
 - data access
 - issues 59
 - data aggregation cycle 84
 - data categories 77
 - data change rate 73
 - data enhancement overlap 84
 - data replication
 - benefits 72
 - business use 88
 - consistency 86
 - data placement 67
 - frequency control 86
 - in profitability analysis 72
 - key interfaces 63
 - objective 83
 - role 61
 - tools 63, 73, 83
 - two-way role 71
 - unit of work information 87
 - data server 97
 - data staging interface 63
 - DataGuide
 - an organization asset data catalog 77
 - DataHub
 - DataPropagator Relational 74, 92
 - Information Warehouse framework 50
 - integration 92
 - interface to tools 92
 - launch tool 62
 - point of control 62, 92
 - task simplification 92
 - DataPropagator NonRelational 87
 - DataPropagator Relational
 - DataHub 92
 - DB2 product matrix 92
 - product 87
 - DB2
 - changed data support 97
 - DataPropagator NonRelational 87
 - DataPropagator Relational 87
 - DB2/6000 30
 - DDCS/2 75
 - distributed unit of work 91
 - family 92
 - I/O parallelism 28
 - load performance 86
 - log structure 99
 - log switching 99
 - propagation matrix 92
 - DB2 DARM 91
 - DDCS/2 75
 - deploying product 57
 - deregulation 15
 - derived data 77
 - distributed data
 - consolidation 12
 - Distributed Relational Database Architecture
 - See DRDA
 - distributing data 71
 - DRDA
 - distributed unit of work 91
 - implementation 75
 - in Access Enablers 59
 - in finance industry 30
 - value 56
- E**
- embedded SQL 58
 - enabling product 57
 - enterprise data model
 - organization asset data representation 47
 - entity
 - business objects 81
 - classify 109
- F**
- fee-based services 66
 - finance enterprise
 - customer base 27
 - finance industry
 - ATM transactions 66
 - attrition rate 18
 - competitive pressures 16
 - currency inventory 70
 - customer role 66
 - expert systems 68
 - global economy 16
 - image 29
 - product marketing 68
 - finance industry study
 - branch organizations 13
 - configuration 75
 - customer demands 13, 15
 - head office 69, 75
 - LAN role 24
 - risk management 13
 - system 74
 - traveler's checks 69, 70

Financial Application Architecture
and IW architecture 40
financial instruments 16
Financial Services Data Model
purpose 47

I

image 15
information
acquisition 18
financial 15
role 12
Information Catalog
finding data 27
information products 23
information systems
decentralized 19
infrastructure 18
reach and range 19
information systems department
mission 12
Information Warehouse
for revenue 24
storage mapping 29
Information Warehouse architecture
and FAA 40
benefits 72
data categories 77
focus areas 60
goals 60
interfaces 73
objective 53
Information Warehouse framework
connectivity 50
DataHub 50
definition 49
enterprise perspective 27
informational application
in profitability analysis 72
informational environment 18
informational object
example 79
insurance industry

K

key systems 20
Information Warehouse architecture and
products 21
knowledge worker
active 18
definition 4

knowledge worker (*continued*)
query systems 28

L

LAN
and large servers 27
in finance 69
work group environment 27
legacy systems 25
loan management 13
loan origination 13

M

market segmentation 14
measurements 14
meta-data
access 58
definition 77
descriptive 18
Information Catalog 77
role 27
tool-specific 96
model
-based application generation 109
abstract 109
data 78
industry focus 79
modeling
MVS/ESA
in finance industry 74

O

object handler interface 63
organization asset data
components 78
enterprise data 77
model 78
organizing 106
representation 47

P

point of control 1
presentation function 1
product diversification 13
product profit 17

profitability analysis 21
 overview 72
propagation
 choosing update or refresh 87
 efficiency 87
 log interface 91
 update 87
pruning 98, 102

Q

query systems 28
quiesce point 87

R

real-time data 77
reconciled data 77
refresh propagation 85
registration 102
replication
 performance 86
 transparency 86
retail industry study
 point-of-sale systems 18
risk analysis 21

S

single point of control 73
solution thread
 objective 4
 requirements 65
SQL call level interface 58
strategy
 finance customer 12
 information systems 17
subscription 102

T

tool invocation interface 63
transaction consistency 100

U

update propagation 86
update semantics 89

update serialization 88
update vs. refresh 86

W

work group platform 27
workflow management
 focus 85
 interface 63

ITSO Technical Bulletin Evaluation

RED000

Information Warehouse in The Finance Industry

Publication No. **GG24-4340-00**

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	_____		
Organization of the book	_____	Grammar/punctuation/spelling	_____
Accuracy of the information	_____	Ease of reading and understanding	_____
Relevance of the information	_____	Ease of finding information	_____
Completeness of the information	_____	Level of technical detail	_____
Value of illustrations	_____	Print quality	_____

Please answer the following questions:

- a) If you are an employee of IBM or its subsidiaries:
Do you provide billable services for 20% or more of your time? Yes____ No____
Are you in a Services Organization? Yes____ No____
- b) Are you working in the USA? Yes____ No____
- c) Was the Bulletin published in time for your needs? Yes____ No____
- d) Did this Bulletin meet your needs? Yes____ No____
- If no, please explain:

What other topics would you like to see in this Bulletin?

What other Technical Bulletins would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

Name

Address

Company or Organization

Phone No.



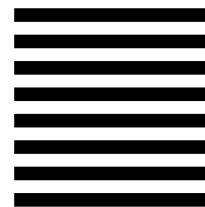
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 471, Building 070B
5600 COTTLE ROAD
SAN JOSE CA
USA 95193-0001



Fold and Tape

Please do not staple

Fold and Tape

