Academic Edition: Applying Pattern Approaches Patterns for e-business Series

Applying Patterns for e-business effectively

Integrating pattern approaches within a methodology

Validating technical scenarios with case studies

Michele Galic
Jonathan Adams
Jon A. Bell
Richard Disney
Ville-Mikko Kanerva
Steve Matulevich
Kent Rebman
Philippe Spaas
Alan Chmura
Dr. Uldarico Rex Dumdum
Jo James

ibm.com/redbooks

Redbooks

International Technical Support Organization

Academic Edition: Applying Pattern Approaches Patterns for e-business Series

November 2007

SG24-7466-00

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (Sept 2007)

This edition applies to Patterns for e-business.

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
CICS®
Domino®
DB2®
DRDA®
e-business on demand®
e-business Hosting™
eServer™
ESCON®
Everyplace®
IBM®
Lotus Notes®
Lotus®
MQSeries®
MVS™
MVS/ESA™
MVS/XA™
Notes®
On Demand Business logo™
OS/390®
Parallel Sysplex®
pSeries®
QuickPlace®
Rational®
Redbooks®
Redbooks (logo) ®
S/390®
Sametime®
Software as Services™
System x™
System z™
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

ABAP, SAP NetWeaver, SAP R/3 Enterprise, mySAP.com, mySAP, SAP R/3, SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Shockwave, and Portable Document Format (PDF) are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Java, Java Naming and Directory Interface, JavaScript, JDBC, JDK, Solaris, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Internet Explorer, Microsoft, PowerPoint, Visio, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Preface

As companies compete in the e-business marketplace, they find that they must re-evaluate their business processes and applications so that their technology is not limited by time, space, organizational boundaries, or territorial borders. They must consider the time it takes to implement the solution as well as the resources (people, money, and time) they have at their disposal to successfully execute the solution. These challenges, coupled with the integration issues of existing systems and the pressure to deliver consistent high-quality service, present a significant undertaking when developing an e-business solution. IBM® Patterns for e-business provide a structured way of decomposing simple or complex requirements into a layered set of reusable architectures and designs.

This IBM Redbooks® publication's focus is on documenting the currently observed usages of the Patterns for e-business. This IBM Redbooks publication is part of the Patterns for e-business series. We introduce three identified pattern approaches and examine pattern scenarios for each of them. The three pattern approaches are:

- Approach 1: Building solutions on the basis of a Single Business pattern
- Approach 2: Building solutions on the basis of Multiple Business patterns
- Approach 3: Applying Patterns with existing Custom designs

We discuss the key features and identify guidelines for each of these approaches. We document the alignment of these approaches within a context of a proven methodology, the IBM Global Services Method.

Introduction for students and instructors

The value of the IBM Patterns for e-business architectures discussed in this IBM Redbooks publication has been proven repeatedly in real life. Our intent in providing this academic edition of the original IBM Redbooks publication *Patterns: Applying Pattern Approaches Patterns for e-business Series*, SG24-6805 is to make these architectures more readily available and understandable to students in institutions of higher learning.

To this end, we have added material not contained in the original IBM Redbooks publication, material intended to reinforce the learning experience for students:

- Learning objectives, which specify expected outcomes for each chapter
- An outline, which emphasizes the major points of each chapter

- Cases, which demonstrate in detail how architectural solutions have proven their value in the real world and suggest approaches students might use in their projects
- Summaries, which restate succinctly the material covered in each chapter
- Test banks, which allow for assessment of learning in fine-grained detail
- Appendices, which offer suggestions for further study

We strongly encourage and endorse the concurrent use of two additional books:

- *Patterns for e-business: A Strategy for Reuse*, by Adams, et al
- *Design Patterns: Elements of Reusable Object-Oriented Software*, by Erich Gamma, et al

Patterns are not new to computing disciplines or business disciplines. Combining both in one coherent package, independent of any development methodology, contributes significantly to the project team's ability to satisfy its mission.

The team that wrote this IBM Redbooks publication

This IBM Redbooks publication was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Michele Galic is a Specialist at the International Technical Support Organization, Raleigh Center. Her focus is on the WebSphere® family of products and Patterns for e-business. She has 13 years of experience in the IT field. She holds a degree in Information Systems. Before joining the ITSO, Michele was a Senior IT Specialist in IBM Global Services in the Northeast, specializing in the WebSphere field.

Jonathan Adams has been an IT architect with IBM for over 30 years. For the last ten years, he has focused on helping IBM deliver reusable end-to-end middleware solutions to its customers. Since September 1998, he has been working in the SWG Solutions & Strategy organization leading the definition and development of Patterns for e-business. These patterns have been built by teams that cross all the major IBM divisions. The resultant patterns are being used by IBMers, Customers, and Business Partners to help reduce risk and increase speed to market on many e-business solution developments.

Jon A. Bell has been in the IT Industry for 23 years. He has worked in the UK, US, Africa, and Australia. Jon's entire IT career has been spent as a technologist. He spent eight years lecturing on subjects ranging from systems development methodologies and relational theory to Business Intelligence and Database performance tuning. Most recently, Jon served five years with Australia's largest telco in their Strategic Architecture group, and for the last two years he has worked in the IBM Software Group's Global Systems Integrators Alliance team as a Solutions Manager. He is a member of the Australian Computer Society, the British Computer Society, and the Computer Society of South Africa.

Richard Disney is an IT Architect and a member of the Architecture and Technology Center of Excellence. He has been with IBM for four years and has eight years of experience in Information Technology. He holds a Bachelor of Science degree from the California State University at Northridge. His areas of expertise include architecture design and development in the Health Care and Insurance Industries using such technologies as XML, DB2® (using the XML Extender functionality), MQ Series, and MQSI. He has also worked extensively with EDI translators to accommodate some of the regulatory requirements of the health care industry.

Ville-Mikko Kanerva is an IT Architect in Helsinki, Finland. He has three years of experience in the telecommunications industry creating portal infrastructures. He holds a Master's degree in Computer Science and Engineering from the Helsinki University of Technology. Ville-Mikko's areas of expertise include internet technologies and protocols, portal infrastructures with an emphasis on authentication, authorization, and accounting, the IBM Global Services Method, and the Java™ programming language.

Steve Matulevich is a Consulting I/T Specialist for the Advanced Technical Support organization in the Washington Systems Center in Gaithersburg, Maryland, USA. His focus is on supporting and promoting e-business solutions on the IBM System z™ platform. He has twenty-three years of experience with IBM, starting as a Computer Operator and a Programmer for IBM Inventory and Billing systems. He spent fourteen years in the IBM Federal Marketing branch office as a Systems Engineer and Marketing Specialist, where he supported federal agencies with large install bases of IBM mainframes and software. During this period, he provided technical support for IBM mainframe operating systems (MVS/XA™, MVS/ESA™, and OS/390®), IBM Storage Subsystems, CICS®, and IBM LAN Network Management products. He currently supports IBM WebSphere products on the System z platform and teaches hands-on workshops to customers who are currently moving their systems to Web-based solutions on the mainframe.

Kent Rebman is a Software IT Architect in Indianapolis, Indiana and has 24 years of experience with IBM in a variety of technical sales support roles, primarily working with customers in the finance and health care industries as a systems engineer. In the early 1990s, he spent several years developing e-business proof of concept systems for a variety of customers and industries. Along with a colleague, he developed and managed all the content, software, and hardware for one of the IBM earliest customer-facing internet sites. His areas of expertise are primarily centered around system design, integration, and architecture, working extensively now with WebSphere, Domino®, and Web services. He holds a psychology degree from Indiana University.

Philippe Spaas is a Consulting IT Architect in Brussels, Belgium. He has ten years of experience working as an application architect on various projects in the financial sector. He participated in several architecture harvesting and method development related projects resulting in key contributions to the Architecture Domain of the IBM Global Services Method and the IBM Architecture Description Standard (ADS). Philippe is actively involved with the education of the IBM architect population both through teaching and course development. He holds a degree in applied economics from the Katholieke Universiteit Leuven (Belgium), and an M.B.A. from Cornell University.

The team that wrote the academic content in this IBM Redbooks publication

Alan Chmura is a writer, consultant, and university faculty member in Cheyenne, Wyoming. He has worked in various capacities on significant development projects in the oil and gas, telecommunication, aerospace, and office equipment industries. He is co-author of a book on logical data modeling, published by Springer in 2005, which extends the literature of object modeling with patterns into the relational database domain, He holds a doctoral degree in Business Administration from the University of Southern California. He is currently researching enterprise architectures.

Dr. Uldarico Rex Dumdum is an associate professor of MIS, e-business, and leadership at Marywood University. His consulting work includes academic program development, technology insertion in organizations, and vision implementation and leadership development. Holds a PhD from the joint program of the Watson School of Engineering and Applied Science and the School of Management at SUNY-Binghamton. Has has done post-graduate study in leadership, has an MBA, and an MS in Computer Science from SUNY-Binghamton and a Master of Engineering from the Asian Institute of Technology in Bangkok, Thailand. Prior to teaching at Marywood, he taught at the School of Management at SUNY-Binghamton in its undergraduate and graduate programs, its Griffiss Air Force graduate program, and its

Lockheed-Martin Executive MBA program. He was a fellow at the Center for Leadership Studies and at the Institute for Student-Centered Learning at SUNY-Binghamton. His awards include Faculty of the Year, Who's Who Among America's Teachers (twice), Best Paper of an IS Education Conference, Fulbright Scholarship, Outstanding Alumnus, nomination for Distinction in One's Discipline, and a Marywood's Favorite Teacher unofficial award from his IT students.

Jo James holds a BS in Management Information Systems from Gardner Webb University, a MBA from Gardner Webb University, a graduate certificate in Web Technology from East Carolina University, and is currently pursuing an Ed. S. in Adult Education from Appalachian University. She currently teaches Computer Programming, Web technology, and Information Systems classes in the Business Sciences division at Isothermal Community College. In addition to her responsibilities as an instructor, Jo serves as a member of the Assessment Taskforce and co-chair of the Learning Strategies Taskforce, which are both part of the Team for the Advancement of the Learning College at Isothermal.

Thanks to the following people for their contributions to this project:

Ian Charters
Senior Consultant, Technology Strategy Consulting (Pan-EMEA Practice), IBM UK

Ed Kahan
Global BIS Solution Architecture and Technology Integration Executive, IBM US

George Galambos
CTO, IBM Canada

Luba Cherbakov
US TI Competency Segment Lead, IBM US

Mark R. Palmer
Global e-business Solutions, IBM US

Ray Harishankar
Architecture & Technology Center of Excellence, IBM US

Leo Marland
Architecture & Technology Center of Excellence, IBM Canada

David H. Janson
Architecture & Technology Center of Excellence, IBM US

Shankar S. Kalyana
Architecture & Technology Center of Excellence, IBM US

Ravi Subramaniam
Architecture & Technology Center of Excellence, IBM US

Mark Calvert
Architecture & Technology Center of Excellence, IBM US

Krishnan K. Ramachandran
Architecture & Technology Center of Excellence, IBM US

Jeanne Tucker
Carla Sadter
International Technical Support Organization, Raleigh Center

Comments welcome

Your comments are important to us!

We want our IBM Redbooks publications to be as helpful as possible. Send us your comments about this or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review IBM Redbooks publication form found at:

  ibm.com/redbooks

- Send your comments in an Internet note to:

  redbook@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Introduction to Patterns for e-business and various pattern approaches

The IBM Patterns for e-business are a collective set of proven architectures that have been compiled from more than 20,000 successful internet-based engagements. This repository of assets can be used by companies to facilitate the development of Web-based applications. They help an organization understand and analyze complex business problems and break them down into smaller, more manageable functions that can then be implemented using low-level design patterns.

Outline

- Introduction to Patterns for e-business
  - The problem
  - The patterns
    - Business patterns
    - Integration patterns
    - Composite patterns
    - Application patterns

        - Runtime patterns
        - Product mappings
        - How the patterns are used
        - Why the patterns are used
- The Patterns for e-business layered asset model
    - The approach
    - Asset definitions
    - Patterns for e-business Web site
- How to use Patterns for e-business
    - Levels of detail
    - The layered structure
    - The primary Business patterns
        - Self-Service (User-to-Business)
        - Collaboration (User-to-User)
        - Information Aggregation (User-to-Data)
        - Extended Enterprise (Business-to-Business)
    - The Integration patterns
        - Access Integration
        - Application Integration
    - Pattern representation of a custom design
    - The Composite patterns
        - Electronic Commerce
        - Portal
        - Account Access
        - Trading Exchange
        - Sell-Side Hub (Supplier)
        - Buy-Side Hub (Purchaser)
    - Selecting the Application pattern, Runtime pattern, and Product mapping
    - Common initiations
        - Executive vision statement
        - Response to an RFP

    - Response to “Line of Business” executive (CRM, SAP®, and WebSphere)
    - Requested technical solution
  - Applying Patterns approaches
  - Single Business pattern
  - Multiple Business patterns
  - Existing Custom designs
  - Patterns for e-Business in software solutioning
    - Solution Outline phase
    - Macro Design phase
    - Micro Design phase
    - Development and testing of the solution
- Structure of this IBM Redbooks publication
  - Intent
  - Structure

Learning objectives

The objectives of this chapter help you to:

- Understand and explain the e-Business Layered Asset Model
- Explain the application of the following pattern types
  - Business
  - Integration
  - Composite
  - Custom

Real-world environment

In this section, we refer you to a real-world situation that uses the concepts in this chapter. The case study, called “Foundation Technology Services cuts time and risk using IBM Patterns” is available on the Web at:

http://www.redbooks.ibm.com/additional_materials/SG247466/Foundation_Technology_Services.pdf

1.1 Introduction to Patterns for e-business

As companies compete in the e-business marketplace, they find that they must re-evaluate their business processes and applications so that their technology is not limited by time, space, organizational boundaries, or territorial borders. They must consider the time it takes to implement the solution, as well as the resources (people, money, and time) they have at their disposal to successfully execute the solution. These challenges, coupled with the integration issues of existing systems and the pressure to deliver consistent high-quality service, present a significant undertaking when developing an e-business solution.

In an effort to alleviate the tasks involved in defining an e-business solution, IBM has built a repository of “patterns” to simplify the effort. In simple terms, a “pattern” can be defined as a model or plan used as a guide in making things. As such, patterns serve to facilitate the development and production of things. Patterns codify the repeatable experience and knowledge of people who have performed similar tasks before. Patterns not only document solutions to common problems, but also point out pitfalls that should be avoided. The IBM Patterns for e-business consist of documented architectural best practices. They define a comprehensive framework of guidelines and techniques that were actually used in creating architectures for customer engagements. The Patterns for e-business bridge the business and IT gap by defining architectural patterns at various levels from Business patterns to Application patterns to Runtime patterns, enabling easy navigation from one level to the next. Each of the patterns (Business, Integration, Application, and Runtime) helps companies understand the true scope of their development project and provide the necessary tools to facilitate the application development process, thereby allowing companies to shorten time to market, reduce risk, and more importantly, realize a more significant return on investment.

The major types of Patterns for e-business are:

- Business patterns
- Integration patterns
- Composite patterns
- Application patterns
- Runtime patterns and matching product mappings

When a company takes advantage of these documented assets, they are able to reduce the time and risk involved in completing a project.

For example, a line-of-business (LOB) executive who understands the business aspects and requirements of a solution can use Business patterns to develop a high-level structure for a solution. Business patterns represent common business problems. A LOB executive can match their requirements (IT and business drivers) to Business patterns that have already been documented. The patterns provide tangible solutions to the most frequently encountered business challenges by identifying common interactions among users, business, and data.

Senior technical executives can utilize Application patterns to make critical decisions related to the structure and architecture of the proposed solution. Application patterns help refine Business patterns so that they can be implemented as computer-based solutions. Technical executives can use these patterns to identify and describe the high-level logical components that are needed to implement the key functions identified in a Business pattern. Each Application pattern would describe the structure (tiers of the application), placement of the data, and the integration (loosely or tightly coupled) of the systems involved.

Finally, solution architects and systems designers can develop a technical architecture by using Runtime patterns to realize the Application patterns. Runtime patterns describe the logical architecture that is required to implement an Application pattern. Solution architects can match Runtime patterns to existing environment and business needs. The Runtime pattern they implement establishes the components needed to support the chosen Application pattern. It defines the logical middleware nodes, their roles, and the interfaces among these nodes in order to meet business requirements. The Runtime pattern documents what must be in place to complete the application but does not specify product brands. Determination of actual products is made in the Product mapping phase of the patterns.

In summary, Patterns for e-business captures e-business approaches that have been tested and proven. By making these approaches available and classifying them into useful categories, LOB executives, planners, architects, and developers can further refine them into useful, tangible guidelines. The patterns and their associated guidelines allow the individual to start with a problem and a vision, find a conceptual pattern that fits this vision, define the necessary functional pieces that the application will need to succeed, and then actually build the application. Furthermore, the Patterns for e-business provides common terminology from a project’s onset and ensures that the application supports business objectives, significantly reducing cost and risk.

1.2 The Patterns for e-business layered asset model

The Patterns for e-business approach enables architects to implement successful e-business solutions through the re-use of components and solution elements from proven successful experiences. The Patterns for e-business approach is based on a set of layered assets that can be exploited by any existing development method. These layered assets are structured in a way that each level of detail builds on the last. These assets include:

- Business patterns that identify the interaction between users, businesses, and data.
- Integration patterns that tie multiple Business patterns together when a solution cannot be provided based on a single Business pattern.
- Composite patterns that represent commonly occurring combinations of Business patterns and Integration patterns.
- Application patterns that provide a conceptual layout describing how the application components and data within a Business pattern or Integration pattern interact.
- Runtime patterns that define the logical middleware structure supporting an Application pattern. Runtime patterns depict the major middleware nodes, their roles, and the interfaces between these nodes.
- Product mappings that identify proven and tested software implementations for each Runtime pattern.
- Best-practice guidelines for design, development, deployment, and management of e-business applications.

These assets and their relationships to each other are shown in Figure 1-1 on page 7.

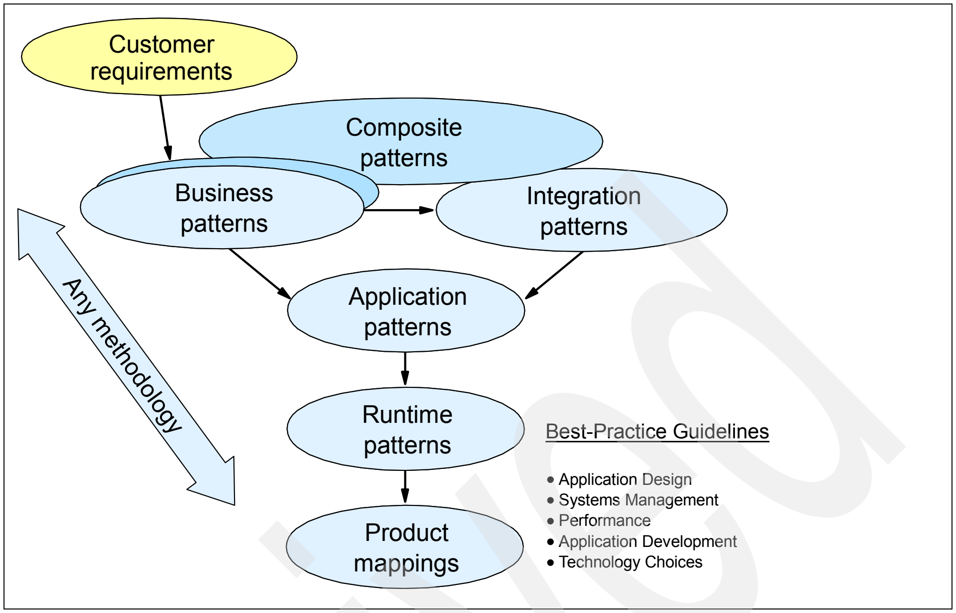


Figure 1-1 The Patterns for e-business layered asset model

1.2.1 Patterns for e-business Web site

The Patterns Web site provides an easy way of navigating top down through the layered Patterns' assets in order to determine the preferred reusable assets for an engagement.

For easy reference to Patterns for e-business, refer to the Patterns for e-business Web site at:

http://www.ibm.com/developerWorks/patterns/

1.3 How to use the Patterns for e-business

As described in the last section, the Patterns for e-business are structured in such a way that each level of detail builds on the last. At the highest level are Business patterns that describe the entities involved in the e-business solution. A Business pattern describes the relationship between the users, the business organization or applications, and the data to be accessed.

Composite patterns appear in the hierarchy (shown in Figure 1-1 on page 7) above the Business patterns. However, Composite patterns are made up of a number of individual Business patterns, and at least one Integration pattern. In this section, we will discuss how to use the layered structure of the Patterns for e-business assets.

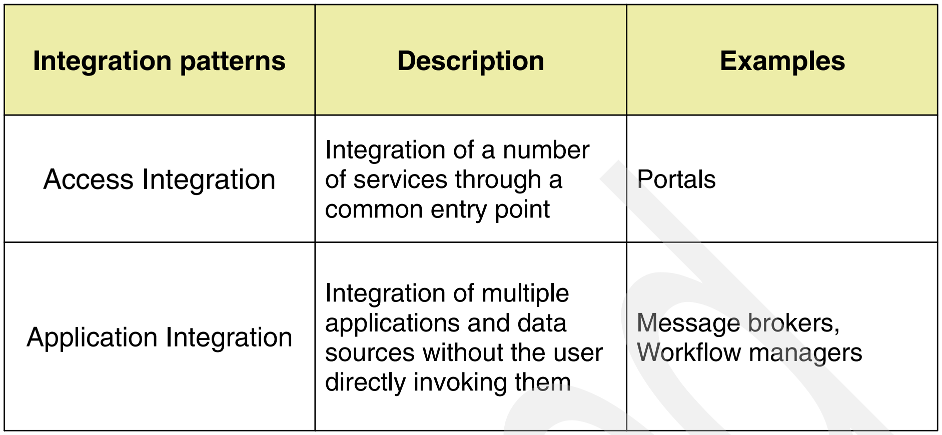
There are four primary Business patterns, as shown in Figure 1-2.

Business patterns	Description	Examples
Self-Service (User-to-Business)	Applications where users interact with a business through the internet and intranet	Simple Web site applications
Collaboration (User-to-User)	Applications where the Internet supports collaborative work between users	E-mail, community, chat, video conferencing, and so on
Information Aggregation (User-to-Data)	Applications where users can extract useful information from large volumes of data, text, images, and so on	Business intelligence, knowledge management, Web crawlers
Extended Enterprise (Business-to-Business)	Applications that link two or more business processes across separate enterprises	EDI, supply chain management, and so on

Figure 1-2 Business patterns

It would be very convenient if all problems fitted nicely into the four Business patterns above, but reality says that things will often be more complicated. The patterns assume that all problems, when broken down into their most basic components, will fit more than one of these patterns. When a problem describes multiple objectives that fit into multiple Business patterns, the Patterns for e-business provide the solution in the form of Integration patterns.

Integration patterns allow us to tie together multiple Business patterns to solve a problem. The Integration patterns are shown in Figure 1-3 on page 9.



Integration patterns	Description	Examples
Access Integration	Integration of a number of services through a common entry point	Portals
Application Integration	Integration of multiple applications and data sources without the user directly invoking them	Message brokers, Workflow managers

Figure 1-3 Integration patterns

These Business and Integration patterns can be combined to implement installation-specific business solutions. We call this a Custom design.

We can represent the use of a Custom design to address a business problem through an iconic representation, as shown in Figure 1-4.

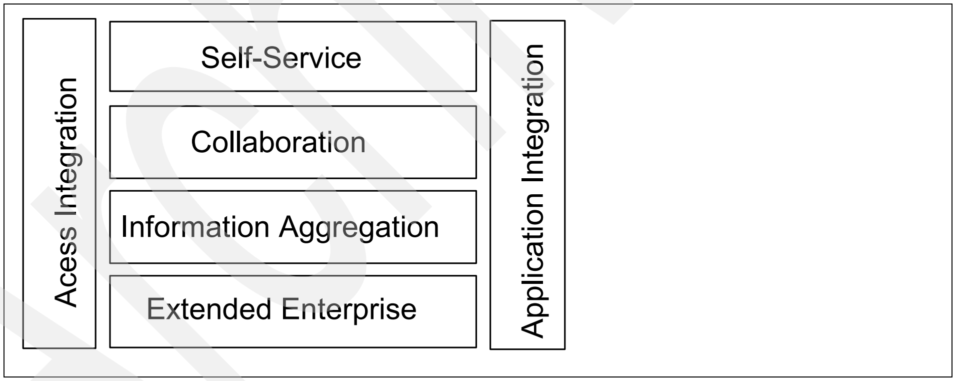


Figure 1-4 Pattern Representation of a Custom design

If any of the Business or Integration patterns are not used in a Custom design, we can show that using blocks lighter than the other blocks. For example, Figure 1-5 shows a Custom design that does not have a mandatory Collaboration business pattern or an Extended Enterprise business pattern for a business problem.

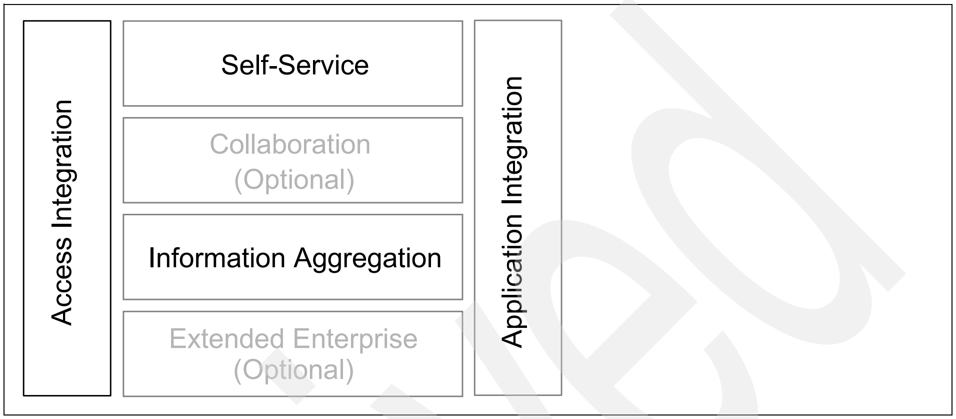
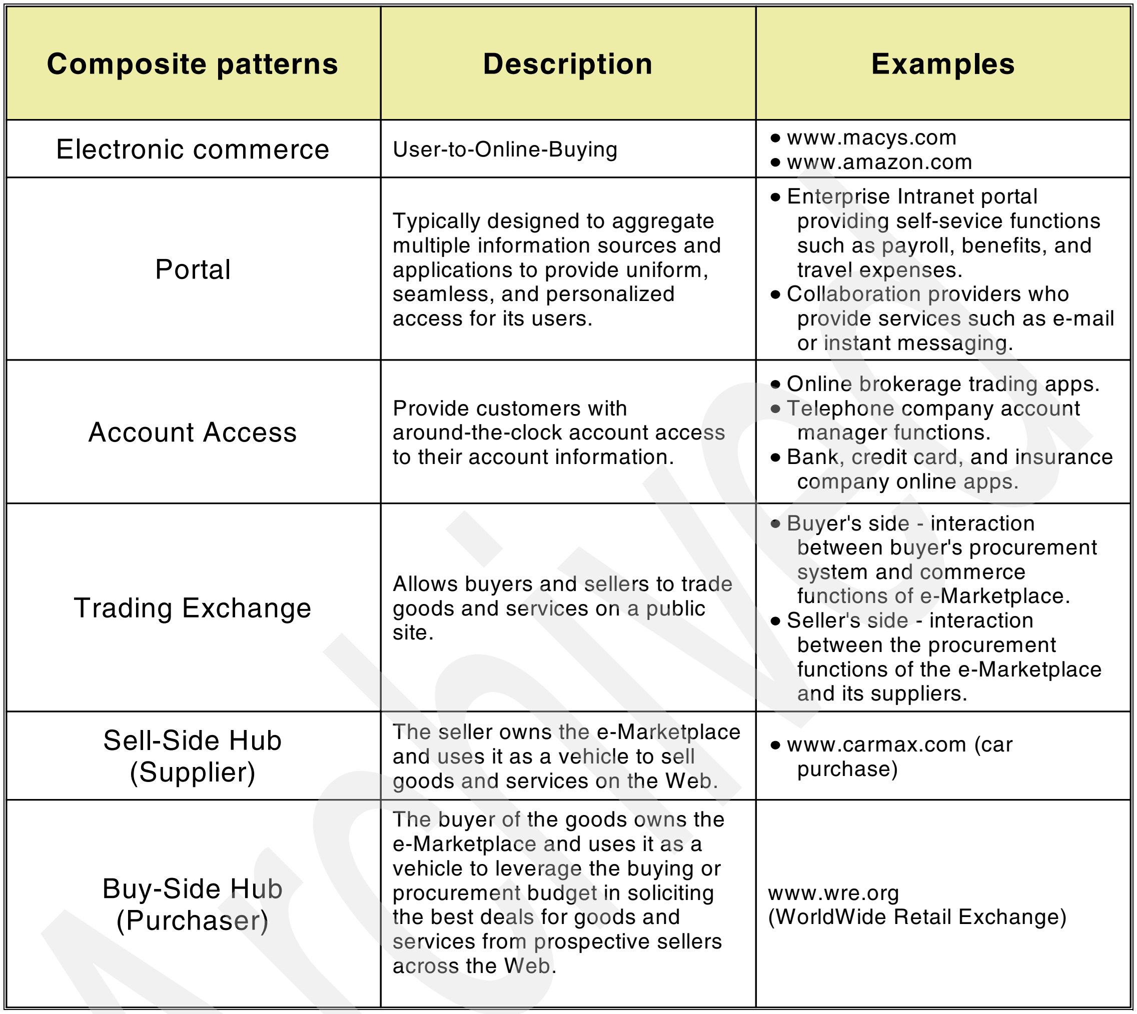


Figure 1-5 Custom design with Self-Service, Information Aggregation, Access Integration and Application Integration

A Custom design may also be a Composite pattern if it recurs many times across domains with similar business problems. For example, the iconic view of a Custom design in Figure 1-5 can also describe a Sell-Side Hub composite pattern.

Several common uses of Business and Integration patterns have been identified and formalized into Composite patterns. The identified Composite patterns are shown in Figure 1-6 on page 11.



Composite patterns	Description	Examples
Electronic commerce	User-to-Online-Buying	• www.macys.com • www.amazon.com
Portal	Typically designed to aggregate multiple information sources and applications to provide uniform, seamless, and personalized access for its users.	• Enterprise Intranet portal providing self-sevice functions such as payroll, benefits, and travel expenses. • Collaboration providers who provide services such as e-mail or instant messaging.
Account Access	Provide customers with around-the-clock account access to their account information.	• Online brokerage trading apps. • Telephone company account manager functions. • Bank, credit card, and insurance company online apps.
Trading Exchange	Allows buyers and sellers to trade goods and services on a public site.	• Buyer's side - interaction between buyer's procurement system and commerce functions of e-Marketplace. • Seller's side - interaction between the procurement functions of the e-Marketplace and its suppliers.
Sell-Side Hub (Supplier)	The seller owns the e-Marketplace and uses it as a vehicle to sell goods and services on the Web.	• www.carmax.com (car purchase)
Buy-Side Hub (Purchaser)	The buyer of the goods owns the e-Marketplace and uses it as a vehicle to leverage the buying or procurement budget in soliciting the best deals for goods and services from prospective sellers across the Web.	www.wre.org (WorldWide Retail Exchange)

Figure 1-6 Composite patterns

The makeup of these patterns is variable in that there will be basic patterns present for each type, but the Composite pattern can easily be extended to meet additional criteria. For more information about Composite patterns, refer to *Patterns for e-business: A Strategy for Reuse by* Adams, et al.

1.3.1 Selecting the Application pattern, the Runtime pattern, and the Product mapping

Once the appropriate Business, Integration, Composite pattern is identified, the next step is to define the high-level logical components that make up the solution and how these components interact. This is known as the Application pattern. A Business, Integration, Composite pattern will usually have multiple Application patterns identified that describe the possible logical components and their interactions. For example, an Application pattern may have logical components that describe a presentation tier for interacting with users, a Web application tier, and a back-end application tier.

The Application pattern requires an underpinning of that middleware that is expressed as one or more Runtime patterns. Runtime patterns define functional nodes that represent middleware functions that must be performed.

Once a Runtime pattern has been identified, the next logical step is to determine the actual product and platform to use for each node. The Patterns for e-business have Product mappings that correlate to the Runtime patterns, describing actual products that have been used to build an e-business solution for this situation.

Finally, guidelines assist you in creating the application using best practices that have been identified through experience.

For more information about determining how to select each of the layered assets, refer to the Patterns for e-business Web site at:

http://www.ibm.com/developerWorks/patterns/

1.3.2 Common initiations

How exactly does one find oneself in a position to execute an architectural methodology or apply the Patterns for e-business? Some common ways are described here. Each of these initiations may include a high level use case.

Executive vision statement

One of the main objectives of every sales professional is to call as high in the customer's organization as possible. In the course of building these executive relationships, they frequently hear statements such as “I want my business partners to have a single company interface.” This is an implicit invitation to spend time understanding multiple parts of the customer organization and uncovering opportunities for software and hardware products to create this vision. This is an opportunity to use the Custom design icon view shown in Figure 1-4 on page 9 to help the executive decompose his/her vision into the required Business and Integration patterns. For example, when decomposing this vision statement, you find that the Self-Service business pattern and the Access Integration pattern are appropriate. This is shown in Figure 1-7.

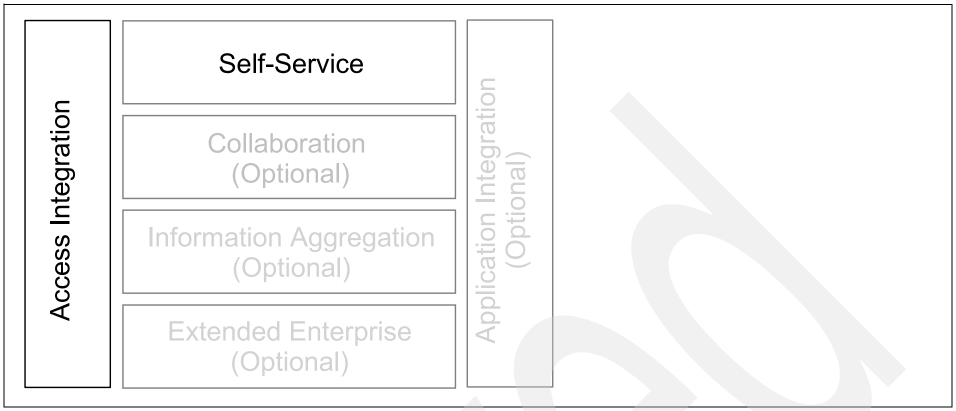

Access Integration
Self-Service
Collaboration (Optional)
Information Aggregation (Optional)
Extended Enterprise (Optional)
Application Integration (Optional)

Figure 1-7 Custom design using the Self-Service business pattern and Access Integration pattern

This Custom design is illustrated in more detail as a case study in Chapter 4, “Building solutions on a multiple Business patterns basis” on page 83.

Response to an RFP

Many customers, primarily in the public sector, use a Request for Proposal (RFP) process. This provides invited vendors with a set of technical requirements, use cases, and evaluation criteria to address already identified customer needs. This also provides information necessary to understand the high level business functions of the proposed solution. Understanding the high level business functions is an excellent starting point to decompose into the customer requirements and identify the necessary Business and Integration patterns. For examples of how the Patterns for e-business contribute to shaping the reply to a RFP, refer to the subsequent chapters in this IBM Redbooks publication.

Response to “Line of Business” executive (CRM, SAP, and WebSphere)

In spending time getting to understand a customer's business, consultants will often hear LOB executives express requirements such as “We must improve the responsiveness of the account inquiry desk by making simple inquiries available to our customers and provide a starting point for future e-business development”. This is an invitation to start investigating a core part of this customer's operation and use the Custom design icon view to decompose this business issue. For example, this business problem can be decomposed into a single Business pattern. The Custom design icon view is shown in Figure 1-8.

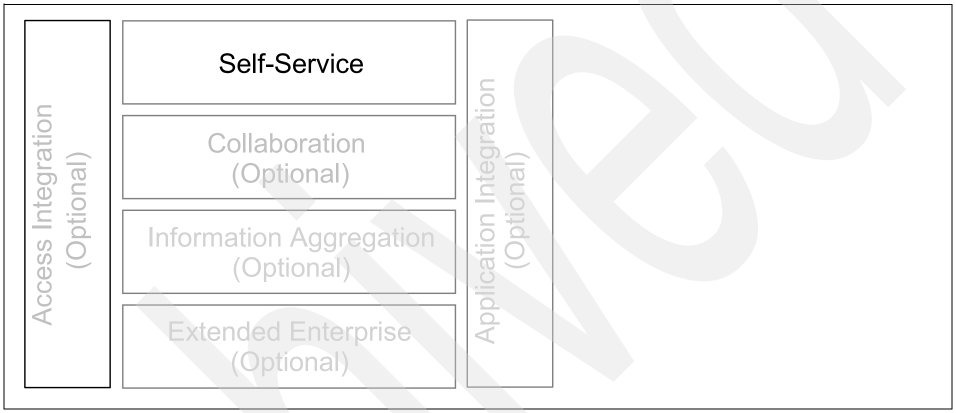

Access Integration (Optional)
Self-Service
Collaboration (Optional)
Information Aggregation (Optional)
Extended Enterprise (Optional)
Application Integration (Optional)

Figure 1-8 Custom design icon view using Self-Service business pattern

This Custom design is illustrated in more detail as a case study in Chapter 3, “Building solutions on a Single Business pattern basis” on page 49.

Requested technical solution

Sometimes, a middleware requirement identifies itself like this: “We need to integrate our WebSphere and Domino systems”. This technical solution is a commonly heard request. In these situations, we may be able to circumvent or shorten many steps in the methodology and move right to the Patterns for e-business using an existing Custom design. For examples of existing Custom designs, refer to Chapter 5, “Applying Patterns with existing Custom designs” on page 137.

1.4 Applying Patterns approaches

Three identified Patterns for e-business approaches will be examined in this IBM Redbooks publication. They are as follows:

- Building solutions on the basis of a Single Business pattern

  Use the patterns for a specific solution based on a single Business pattern.

- Building solutions on the basis of Multiple Business Patterns

  Implement a single coarse-grained application associated with one high-level use case using a Custom design employing multiple Business and Integration patterns.

- Applying Patterns with existing Custom designs (for example, WebSphere and Domino integration)

  The existing Custom design solutions, like Composite patterns, combine Business and Integration patterns to create advanced, end-to-end e-business applications. These solutions, however, have not been implemented to the extent of the Composite patterns. While some of the existing Custom designs that we discuss have been proven, their use has been in a limited number of engagements and they are not yet defined as a Composite pattern.

This IBM Redbooks publication will provide the reader with usage recommendations, case studies for each of the approaches, as well as the alignment of these approaches within a methodology framework. Many IT consulting firms have developed and proven methodologies. It is worth noting that the Patterns for e-business can be exploited within any comprehensive application development methodology. This IBM Redbooks publication will demonstrate such an approach and illustrate how the Patterns can be applied to the development phases within the IBM Global Services Method.

1.4.1 Patterns for e-business in software solutioning

By using proven methodologies in conjunction with Patterns for e-business, you will be able to develop a solution that uses common terminology from the onset of the project. Patterns for e-business can be easily aligned to existing application development life cycles when addressing a complex business goal or requirement. Patterns for e-business are not a software methodology. Patterns for e-business complement any known software methodology. For example:

- Solution Outline phase of a project: An organization can quickly infer the overall shape of the required solution by matching its Business and IT goals/objectives to the Patterns for e-business architectural assets and follow the navigation through the asset catalogue to derive a first cut definition of a solution meeting their business requirements This approach will allow the architect the focus on the most challenging areas of the proposed solution by concentrating on the architecturally significant requirements.

- Macro Design phase of a project: Based on all requirements, the results arrived at during the Solution Outline phase of the project can be validated and further refined leading towards the definition of the target solution based on the relevant Application, Integration, and Runtime patterns and taking into account the Product mapping suggested by Patterns for e-business.
- Micro Design phase: The results delivered at the end of the Macro Design are further detailed taking into account the technology and product selection made previously in order to prepare for the subsequent Build phases.
- Development and testing of the solution: One can leverage more than twenty IBM Redbooks publications that have been published under the IBM Patterns for e-business series.

Through re-use of these proven approaches, IT architects, in 80% of their cases, will be able to analyze a business problem and quickly develop a required solution to address it.

1.5 Structure of this IBM Redbooks publication

The intent of this IBM Redbooks publication is twofold:

1. Education on the different pattern approaches
2. Examples based on roles

Based on your primary role and interest in the IBM Redbooks publication, you may find that you read all the chapters for education on the different pattern approaches or you may read specific chapters. You may read each chapter independently of the others if you are solely interested in the approach of a single business solution, for example. We encourage you to take a look at the table of contents for help in deciding which chapters to read. However, each of the chapters includes information that every reader will be interested in and be able to apply to their job for a more rounded picture. A Patterns glossary has also been added to this IBM Redbooks publication with definitions of the terminology used.

The remainder of this IBM Redbooks publication is structured with the following chapters:

- Chapter 2, “Business Solution Delivery methods” on page 27

  This chapter describes the IBM Global Services Method (The Method).

- Chapter 3, “Building solutions on a Single Business pattern basis” on page 49

  This chapter describes a case study leveraging a single Business pattern using the IBM Global Services e-business Custom engagement model.
- Chapter 4, “Building solutions on a multiple Business patterns basis” on page 83

  This chapter describes a case study using a case study leveraging multiple Business patterns using the IBM Global Services e-business Custom engagement model.
- Chapter 5, “Applying Patterns with existing Custom designs” on page 137

  This chapter covers three existing Custom designs.

Experience with more than 20,000 internet-based engagements has yielded a set of proven architectures (called patterns) that can be used to facilitate the development of Web-based applications. Using high-level patterns, complex business problems can be decomposed into more manageable functions that can be implemented using lower-level design patterns. The proper application of both the high-level patterns and the lower-level patterns can contribute to development projects that are more likely to be delivered on budget, on time, with reduced risk, and with the desired functionality.

- The high-level patterns are:
  - Business patterns
  - Integration patterns
  - Composite patterns
- The lower-level patterns are:
  - Application patterns
  - Runtime patterns
  - Product mappings
- The IBM repository of patterns provides guidance with respect to:
  - Best practices, including proven guidelines and techniques
  - Solutions to common problems
  - Pitfalls to avoid

1.5.1 Key terms and concepts

Please refer to “Glossary” on page 203 for a list of key terms and concepts.

1.6 Questions

Answer the following questions. To check your answers, see "Chapter 1: Answers to questions" on page 170.

1.6.1 Discussion and review questions

1. IBM has built a repository of patterns to simplify the efforts needed to define an e-business solution. What is a pattern and what tasks are simplified or accomplished using such a model?
2. What assistance is provided by the use of the patterns identified in this chapter?
3. What is meant by the layered asset model? How do the assets identified in the Patterns for e-business model relate to one another? What are the assets identified in this chapter?
4. What is the difference between a Custom design and a Composite pattern?
5. How does one find oneself in a position to execute an architectural methodology or apply the Patterns for e-business?

1.7 Exercise and challenge questions

Complete the following exercises and challenge questions for further understanding.

1.7.1 Challenge questions

1. Visit the Patterns for e-business Web site at:

   `http://www-128.ibm.com/developerworks/patterns/`

   What are the four premier Business patterns identified?
2. Visit the Patterns for e-business Web site at:

   `http://www-128.ibm.com/developerworks/patterns/.`

   What are the four premier Composite patterns identified?

   How does this differ from the information provided in the text?

1.7.2 Team project

The three-fold objective of the team project is as follows:

1. To explore the extensive resources for Patterns for e-business on the Web.

   The Patterns Resource page provides links to Patterns development resources: IBM Redbooks publications, white papers, Screen Cams, Web casts, customer references, and the zipped translated Web sites. All these resources are now centrally located on the Patterns Resources page for easy access at the following URLs:

   http://www-128.ibm.com/developerworks/patterns/
   http://www-128.ibm.com/developerworks/patterns/library/index.html

2. To gain an overview of the Patterns for e-business, do the following:

   In the Patterns Resources page:

   http://www-128.ibm.com/developerworks/patterns/library/index.html

   a. Click **Flash Tutorial** for a Patterns Overview Tutorial. The Flash tutorial to the Patterns for e-business is located at:

      http://www-128.ibm.com/developerworks/patterns/multimedia/version4/Flash_web.swf

   b. Click **Take the Journey** Volumes 1, 2, and 3 for a Screen Cam Presentation by Jonathan Adams, SWG Solutions and Strategy.

      Screen Cams Presenting the updated Patterns Web site and solution designs include:

      - Volume 1, ZIP 4770 KB
      - Volume 2, ZIP 5940 KB
      - Volume 3, ZIP 6990 KB

3. To gain a better understanding of the rationale for the development of Patterns for e-business, refer to the following resources in Appendix B, “Resources” on page 197:

   a. *Patterns of Experience: A Review of IBM Patterns for e-Business Initiative*, found at:

      http://www-128.ibm.com/developerworks/patterns/library/index.html

   b. *Patterns for e-business - continued evolution*, found at:

      http://www-128.ibm.com/developerworks/patterns/library/index.html

   c. Component Development Strategies Monthly Newsletter, found at:

      http://www-128.ibm.com/developerworks/patterns/library/index.html

d. *Facilitating the application development process using the IBM Patterns for e-business*, found at:

   http://www-128.ibm.com/developerworks/patterns/library/index.html

Team project writing assignment and presentation for class discussion: In a team of three or four, write a ten- to twelve-page paper that (a) discusses the rationale for the development of Patterns for e-business; and (b) characterizes the Patterns. In addition, your team is to prepare a set of PowerPoint® slides for presentation and class discussion purposes.

1.7.3 Multiple choice

1. As companies compete in the e-business marketplace, they find that they must re-evaluate their business process and applications so that their technology is not:

   a. Limited by time and space.

   b. Limited by organizational boundaries.

   c. Limited by territorial boundaries.

   d. All of the above.

2. Use of the patterns process allows the company to:

   a. Reduce risk involved in project.

   b. Increase time involved in project.

   c. Both A and B are correct.

   d. Neither A or B is correct.

3. ___ represent tangible solutions to the most common business challenges by identifying common interactions between users, business, and data.

   a. Business patterns.

   b. Application patterns.

   c. Runtime patterns.

   d. None of the above.

4. ____ are used to identify and describe the high-level logical component to implement the key functions identified in the Business pattern.

   a. Integration patterns.

   b. Application patterns.

   c. Runtime patterns.

   d. None of the above.

5. Solution architects develop a technical architecture by using _____, which describe the logical architecture that is required to implement an Application pattern.

   a. Business patterns.

   b. Integration patterns.

   c. Runtime patterns.

   d. None of the above.

6. ___________ represent commonly occurring combinations of Business patterns and Integration patterns.

   a. Runtime patterns.

   b. Composite patterns.

   c. Application patterns.

   d. None of the above.

7. _____ provide a conceptual layout describing how the application components and data within a Business pattern or Integration pattern interact.

   a. Runtime patterns.

   b. Composite patterns.

   c. Application patterns.

   d. None of the above.

8. _____ depict the major middleware nodes, their roles, and the interfaces between these nodes.

   a. Runtime patterns.

   b. Composite patterns.

   c. Application patterns.

   d. None of the above.

9. The Business pattern called _____ is an application in which the users interact with a business through the internet and intranet.

   a. Collaboration.

   b. Self-Service.

   c. Information Aggregation.

   d. None of the above.

10. The Business pattern called _____ is an application in which the internet supports collaborative work between users.

    a. Collaboration.

    b. Self-Service.

    c. Information Aggregation.

    d. None of the above.

11. The Business pattern called _____ is an application in which users can extract useful information from large volumes of data, text, images, and so on.

    a. Collaboration.

    b. Self-Service.

    c. Information Aggregation.

    d. None of the above.

12. The Business pattern called _____ is an application that links two or more business processes across separate enterprises.

    a. Collaboration.

    b. Self-Service.

    c. Information Aggregation.

    d. None of the above.

13. The Integration pattern called _____ is the integration of a number of services through a common entry point.

    a. Access Integration.

    b. Application Integration.

    c. Both A and B are correct.

    d. Neither A or B is correct.

14. The Integration pattern called _____ is an integration of multiple applications and data sources without the user directly invoking them.

    a. Access Integration.

    b. Application Integration.

    c. Both A and B are correct.

    d. Neither A or B is correct.

15. A Custom design may also be a(n) _____ if it recurs many times across domains with similar business problems.

   a. Runtime pattern.

   b. Composite pattern.

   c. Application pattern.

   d. None of the above.

16. The Composite pattern called _____ is typically designed to aggregate multiple information sources and applications to provide uniform, seamless, and personalized access for its users.

   a. Trading Exchange.

   b. Portal.

   c. Account access.

   d. None of the above.

17. The Composite pattern called _____ provides customers with around-the-clock account access to their account information.

   a. Trading Exchange.

   b. Portal.

   c. Account access.

   d. None of the above.

18. The Composite pattern called _____ allows buyers and sellers to trade goods and services on a public site.

   a. Trading Exchange.

   b. Portal.

   c. Account access.

   d. None of the above.

19. The Patterns for e-business have Product mappings that correlate to the _____ , describing actual products that have been used to build an e-business solution for this situation.

   a. Runtime patterns.

   b. Composite patterns.

   c. Application patterns.

   d. None of the above.

20. How does one find oneself in a position to execute an architectural methodology or apply the Patterns for e-business?

    a. Executive Vision statement.

    b. Response to an RFP.

    c. Requested technical solution.

    d. All of the above.

21. Through re-use of proven IT approaches identified in the chapter, IT architects, in _____ of cases, will be able to analyze a business problem and quickly develop a required solution to address the problem.

    a. 60%.

    b. 70%.

    c. 80%.

    d. 90%.

1.7.4 Fill in the blank

1. ____________________________ consist of documented architectural best practices.

2. ______________ capture(s) e-business approaches that have been tested and proved.

3. The Patterns for e-business approach is based on a set of ___________________ that can be exploited by any existing development method.

4. __________________ tie multiple Business patterns together when a solution cannot be provided based on a single Business pattern.

5. ________________________ identify proven and tested software implementations for each Runtime pattern.

6. The Business pattern called Self-Service is also known as the _____________________ pattern.

7. The Business pattern called Information Aggregation is also known as the ____________________________ pattern.

8. Business and Integration patterns can be combined to implement an installation-specific business solution. This solution is called a _____________________.

9. With the Composite pattern ____________________ , the seller owns the e-Marketplace and uses it as a vehicle to sell goods and services on the Web.

1.7.5 True or false

1. Patterns not only document solutions to common problems, but also point out pitfalls that should be avoided.
2. The layered asset model layers assets in such a way that each level of detail builds on prior levels.
3. The highest level of the e-business Patterns model is Runtime patterns, which describe the entities involved in the e-business solution.
4. All problems fit into one of the four Business patterns identified in the chapter.
5. In the Buy-Side Hub (Purchaser) Composite pattern, the buyer of the goods owns the e-Marketplace and uses it as a vehicle to leverage the buying or procurement budget in soliciting the best deals for goods and services from prospective sellers across the Web.
6. The makeup of Composite patterns is fixed for the basic types and cannot be easily extended to meet additional criteria.
7. A Business, Integration, or Composite pattern always has a single Application pattern identified that describes the logical components and their interactions.
8. Patterns for e-business are not a software methodology; rather, it complements any known software methodology.

1.7.6 Short answer

1. What is a pattern and what tasks are simplified or accomplished using such a model?
2. How does one find oneself in a position to execute an architectural methodology or apply the Patterns for e-business?

2

Business Solution Delivery methods

This chapter describes the iterative and incremental Business Solution Delivery methods used to build an e-business application starting from requirements gathering and ending with deployment. Many projects rely on mature methods for successfully delivering business solutions. In these methods, the definition of the overall architecture is one of the critical activities. It is precisely to these activities that the Patterns for e-business make their most significant contribution.

In subsequent chapters, we discuss how Patterns for e-business effectively contribute to the definition and delivery of business solutions using several case studies. Throughout these case studies, the IBM Global Services Method will be used as the overall context within which the Patterns for e-business related activities take place. But in principle, any other delivery method could be used to provide such a context.

This chapter will introduce the key concepts of the IBM Global Services Method and explain the phase/activity structure that is used by the Method to structure the overall delivery of a business solution and serves as the backdrop to the Patterns for e-business related activities.

Outline

- Introduction to IBM Global Services Method
  - Work Products and Work Product Descriptions
    - Work Products
    - Work Product Description
  - Method Families and Engagement Models
    - Engagement Models
    - Work Breakdown Structure
    - Relevance of the Global Services Method for proposal development
    - Introduction to specific engagement models
    - e-business Custom
    - e-business Design
    - e-business Express
    - e-business Strategy
- e-business Custom engagement model
  - Solution Startup phase
  - solution Outline phase
  - Macro Design phase
  - Micro Design phase
  - Build Cycle phase
  - Deployment phase
  - Solution Close phase

Learning objectives

The objectives of this chapter help you to:

- Understand and explain the IBM Global Services Method
- Explain Work Product Descriptions for each of the six domains:
  - Engagement
  - Business
  - Organization
  - Application
  - Architecture

  - Operations
- Explain the phases of the Custom Engagement Model
  - Solution Startup
  - Solution Outline
  - Macro Design
  - Micro Design
  - Build Cycle
  - Deployment
  - Solution Close

Real-world environment

In this section, we refer you to a real-world situation that uses the concepts in this chapter. The case study, called “Mohawk Industries keep its customers covered with e-business solutions from IBM”, is available on the Web at:

```
http://www.redbooks.ibm.com/additional_materials/SG247466/Mohawk_Industries.pdf
```

2.1 Introduction to the IBM Global Services Method

The IBM Global Services Method provides a comprehensive body of knowledge allowing practitioners to conduct engagements in a standardized fashion by relying on predefined engagement models and delivering the overall business solution through the creation of a number of well defined work products. As such, the Global Services Method is one of the cornerstones supporting IBM Global Services' goal of shifting its business model towards the delivery of asset-based services by providing a mechanism for practitioners to systematically reuse knowledge and assets using a consistent and integrated approach.

We will briefly introduce a number of key concepts defined within the IBM Global Services Method before discussing each one of these in more detail.

The IBM Global Services Method itself is organized around a number of method families. Method families reflect the close relationships between a number of engagement models applicable to the same domain such as e-business (grouped in the e-business method family), application maintenance (grouped in the application maintenance services method family), and so on.

Engagement models describe typical approaches in delivering a project in terms of a work breakdown structure to be adhered to, the work products to be created, the techniques to be applied, and the roles (in terms of skills) to be allocated to the project.

The work breakdown structure describes the sequence of phases, activities, and tasks to be performed in order to deliver the business solution. During these tasks, one or more work products are created or updated using one of the available techniques in the Method.

Work products can refer to any tangible asset produced during an engagement.

It should be noted that defining the actual engagement model to be used on a project is one of the key activities of the initial phase of any project. During this activity, referred to as *method tailoring*, the actual phases, activities, and tasks will be defined, as well as the relevant work products and techniques. The need for method tailoring underlines the fact that the IBM Global Services Method caters to many different types of engagements and will always require customization to make it applicable to a specific project.

2.1.1 Work products and work product descriptions

Work products (WP) are one of the key elements of the IBM Global Services Method. Work products are tangible artifacts that are produced during a project. They include models, reports, diagrams, plans, code, and other documents that are direct stepping stones towards the delivery of the business solution. They have a specific purpose in the engagement and describe specific content using predefined semantics and syntax. Work products are produced as a result of performing one or more tasks. Some tasks produce less tangible outputs (for example, pass/fail, trained students, and so on) that are then called *outcomes*.

Work products are not the same as deliverables. The term *deliverable* is used to refer to a document that contractually needs to be delivered to the customer, which is not necessarily the case for work products. Some WPs are effectively used as intermediate artifacts and are not intended to be delivered to the customer. All deliverables are created from work products, but not all work products are deliverables.

A Work Product Description (WPD) describes a particular type of work product and gives guidance on the following:

- How to create the actual work product
- Notation
- The recommended structure of the work product itself

The WPDs cover a broad range of work performed on engagements and have been further organized into a number of domains (six in total) and subdomains to further categorize content where needed. These WPDs are available to all engagement models. The six high-level domains are as follows:

- Engagement: Includes those WPDs used to define, plan, manage, and close an IBM Global Services Method engagement.
- Business: Provides the WPDs needed to conduct a structured investigation of the current and desired situations in a customer's business in order to develop and implement the appropriate solution.
- Organization: Contains the WPDs used to assess organizational and cultural impacts and issues, in order to determine the customer's ability to absorb the change required to develop and implement a new solution.
- Application: Provides WPDs for the full life cycle of application development, including design, development, testing, and maintenance of applications.
- Architecture: Contains the WPDs used to structure a solution architecture, including the hardware and software components, the externally visible interfaces, and the relationships among them.

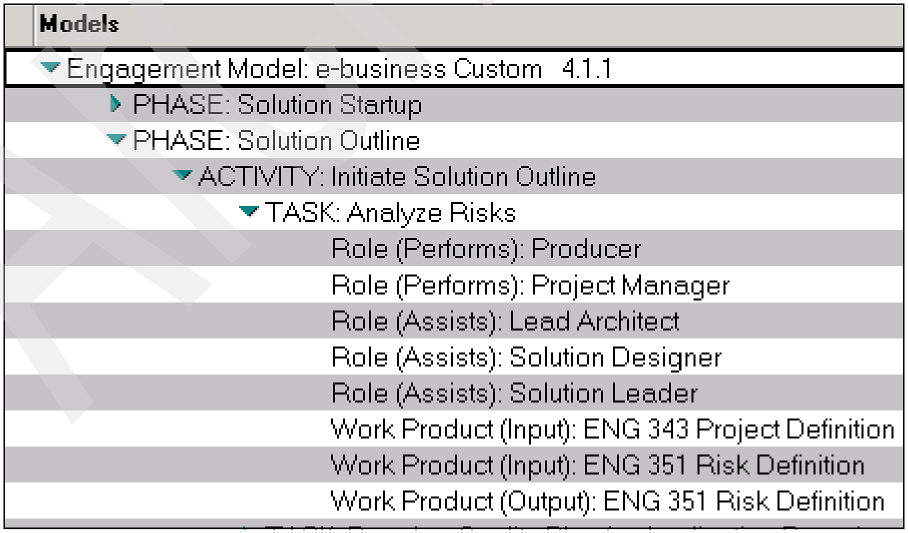
- Operations: Includes the WPDs pertaining to the ongoing operations of a solution, including facilities management, problem reporting, information technology management, and security considerations

2.1.2 Method families and Engagement Models

In addition to the Work Product Descriptions, the Global Services Method also provides guidance on how engagements should be conducted. This guidance is delivered through a number of Engagement Models that represent how typical projects are conducted. These Engagement Models have themselves been further organized into seven method families.

Engagement models provide guidance on the required phases, activities, and tasks (often referred to as the Work Breakdown Structure (WBS)), the Work Products to be produced, the roles to be assigned to the project, and any applicable techniques.

A Work Breakdown Structure (WBS) is a default, skeleton plan. It breaks down the work of the project into a hierarchical structure that includes the phases of the engagement model and the aforementioned activities and tasks. A WBS decomposes what must be done to fulfill the objectives of the project into progressively smaller and more manageable units of work. The WBS describes dependencies between work items and establishes various checkpoints to control the work. With the dimensions of time and resources added, the WBS can be transformed into a project plan. An example of the partial WBS for the e-business Custom engagement model is shown in Figure 2-1.



Models

- Engagement Model: e-business Custom 4.1.1
  - PHASE: Solution Startup
  - PHASE: Solution Outline
    - ACTIVITY: Initiate Solution Outline
      - TASK: Analyze Risks
        - Role (Performs): Producer
        - Role (Performs): Project Manager
        - Role (Assists): Lead Architect
        - Role (Assists): Solution Designer
        - Role (Assists): Solution Leader
        - Work Product (Input): ENG 343 Project Definition
        - Work Product (Input): ENG 351 Risk Definition
        - Work Product (Output): ENG 351 Risk Definition

Figure 2-1 Example of a Work Breakdown Structure for initiate solution outline

As you can see from Figure 2-1 on page 32, the WBS gives a complete description of a task embedded within its phase/activity context. Throughout the engagement model, many tasks, roles, and work products appear multiple times, illustrating that many work products are revisited and further completed during the running of the project that in itself underlines the iterative and incremental nature of the approach supported by the method.

Relevance of the Global Services Method for proposal development

The knowledge embodied in the Global Services Method under the form of the Work Breakdown Structure (WBS), work product descriptions, and techniques, is not only useful during the delivery of a project, but also during the development of a project proposal. This effort, typically limited in time, can also be organized around a subset of the WBS defined in the Method. The definition of the WBS work products to adopt and techniques to apply would take place during the proposal-specific method tailoring activity. The domain structure and Work Product Descriptions of the Method can also help to ensure adequate coverage of the problem domain and help to organize the definition of key aspects of the proposed solution.

One of the key benefits of working during proposal development with a subset of the work products relevant to the solution delivery project is that the proposed solution is already expressed in terms of the work products that will be used by the delivery team.

To illustrate this point, we have structured some of the case studies presented in the following chapters to clearly show how the initial work undertaken during proposal development can feed into the work undertaken during solution delivery.

Introduction to specific engagement models

The IBM Global Services Method Release 4.0 catalog currently details some twenty method families, some of which are listed below:

- Application Management Services
- Business Intelligence
- Custom Application Development
- e-Business
- e-Procurement
- Enterprise Architecture Consulting
- Packaged Solution Integration

For the purposes of this chapter, we will discuss one method family: e-Business.

The e-Business method family consists of the following engagement models:

- e-business Custom
- e-business Design
- e-business Express
- e-business Strategy

For the purposes of providing a business solution delivery method context to our case studies, we will further detail one engagement model: the e-business Custom engagement model from the e-Business method family.

2.2 e-business Custom engagement model

The e-business Custom engagement model is intended for medium to large custom, media-rich, interactive application development projects with a major online or new media delivery aspect: internet, intranet, extranet, CD-ROM, DVD, DVD-ROM, kiosk, and so on. This model emphasizes the importance of creating a compelling user experience that extends a company's brand identity into the new medium.

Special attention is paid to the design and implementation of a large number of new media elements: audio, video, motion graphics, still graphics, copy, and so on. The majority of the new source code produced is either media platform based (Shockwave®, Flash, HTML, DHTML, Java-script, Perl, CGI, and so on) or object-oriented (Java, C++, Smalltalk, and so on).

The overall phasing of an e-business Custom engagement model is shown in Figure 2-2 on page 35:

- Solution Startup: Covers the required start activities.
- Solution Outline: Provides the client with the necessary costs, schedule, and risk information to make an informed investment decision regarding a potential new system.
- Macro Design: Develops a robust architectural framework upon which to build agile releases.
- Micro Design: Prepares for the build cycle of a specific release of the system by driving the architecture and design to a release- and implementation-specific view.
- Build Cycle: Incrementally develops and tests the system until the objectives of the release are achieved.
- Deployment: Deploys the system and prepares for the next release of the system.

- Solution Close: Covers the required close activities.

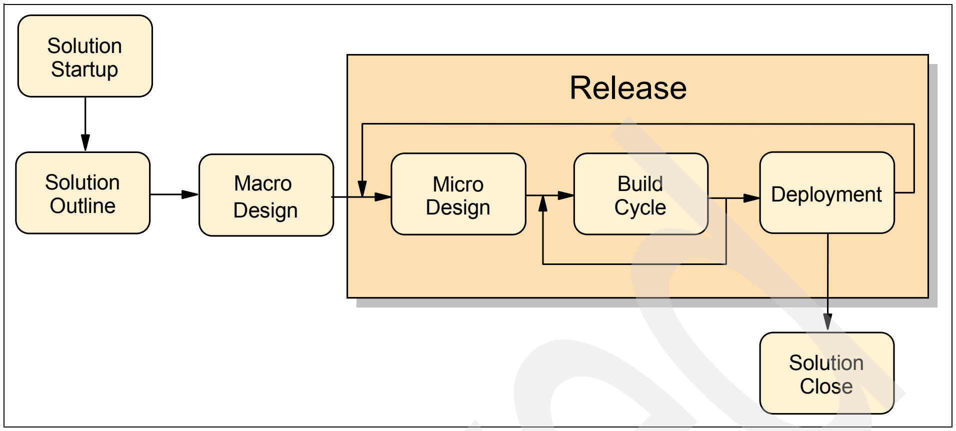


Figure 2-2 e-business Custom Engagement Model process

As shown in Figure 2-2, the Solution Outline and Macro Design phases are only conducted once, whereas the three phases that jointly make up a Release can be performed many times, each time resulting in a release of the system. The Solution Startup and Solution Close phases bracket the overall engagement.

The following sections will discuss these phases in more detail.

2.2.1 Solution Startup phase

The Solution Startup phase focuses on the activities and tasks needed to define, plan, and start an engagement. It is important to carefully define and plan a project up front. All key stakeholders must clearly understand and support the scope and objectives of the project. This will allow you to identify risks that could negatively impact the project and the team's ability to deliver.

The main focus of the activities during this phase is to tailor the engagement model in order to determine the actual phase/activity/task structure that will be followed during the project and the work products that will be created throughout the project. Once these decisions have been taken, the actual project plan detailing all phases/activities/tasks and their associated resources and timing can be defined.

2.2.2 Solution Outline phase

During the Solution Outline phase, the high level business requirements, business objectives, and IT objectives are taken into consideration, focusing each time on those requirements/objectives influencing the overall architecture. Based on these requirements, an overall architecture is defined (the solution outline). The primary objective of this phase is to determine the complexity and scope of the solution. If we understand the complexity and scope, we are able to provide the client with the necessary information to make the investment decision.

Several activities are carried out during the Solution Outline phase, each with the objective of determining the main requirements (functional as well as non-functional) and formulating an initial approach to tackle these requirements. Multiple areas within the requirements domain may be considered, such as:

- Current IT environment
- Security
- User experience
- Business impact

Some of these areas may not be applicable, depending on the scope of the given project.

2.2.3 Macro Design phase

During Macro Design, the overall architecture of the IT system is specified, which allows one to define solution-wide approaches for addressing both the functional and non-functional requirements. This overall architecture acts as a frame of reference for each of the subsequent releases that will deliver (parts of) the architecture so that the release-specific objectives can be achieved. Also, the more detailed work carried out during these releases will be used to refine the overall architecture.

In general, it is far better to spend time up front in Macro Design to address the design issues that impact every release (global decisions) than to rush off to deliver a particular release. The standards' guidelines and architectural work products developed in Macro Design will minimize long term development costs by eliminating as much rework as possible in going from one release to the next.

The Macro Design phase groups activities, moving the work forward that was started during the Solution Outline. The main objective is to confirm the approach and decisions taken during that phase. Typically, the approach defined during the Solution Outline is based on the analysis of the architecturally significant functional and non-functional requirements, and will now be validated by considering all requirements.

2.2.4 Micro Design phase

The purpose of the Micro Design phase is to prepare for the Build Cycle of a specific release of the system by driving the architecture and design to a release and implementation specific view.

A number of activities take place during this phase, including:

- The specification of the required training and user support for a particular release.
- The refinement of the architecture based on the technology and product selection.
- The execution of the static testing processes.

2.2.5 Build Cycle phase

The purpose of the Build Cycle phase is to incrementally develop and test part of the system until the objectives of the overall release are achieved. Releases are defined as increments of the overall solution that are effectively made available to the customer, each release consisting of a Micro Design phase, one or more Build Cycles, and a Deployment phase.

Earlier phases of the Method address the requirements and outline the solution based on these requirements. In a complex solution involving several aspects of the business, the overall solution can be broken down and delivered in smaller increments, releases, each with their own objectives. Within a specific release, multiple Build Cycles could be planned, each with their own objectives in line with and contributing to the overall Release specific objectives. Such a Build Cycle would then consist of a single programming and user experience implementation cycle. In this way, an iterative and incremental implementation approach is adopted that delivers more and more functionality of the overall system with each release/cycle.

2.2.6 Deployment phase

The purpose of this phase is to deploy the system and prepare for the next release of the system.

A number of activities are included in this phase, covering areas such as:

- The setup of the production environment
- The execution of the acceptance tests
- The cut over to production

2.2.7 Solution Close phase

The Solution Close phase is a formal closure to the entire project. During this phase, the results of the project are assembled, analyzed, and archived for use by other practitioners. Solution Startup and Solution Close are the bookends to every engagement. Just as Solution Startup is largely an introduction to the project, Solution Close is essentially a wrap-up.

2.3 Conclusion

A sound and mature Business Solution Delivery method is one of the essential ingredients of any approach for delivering effective solutions. In practice, the Patterns for e-business can make an important contribution to the definition of (parts of) the overall solution. This contribution is typically made during the architecture definition related activities. Therefore, as we step through the remaining chapters of this book, we will be focusing on defining e-business solutions addressing real business cases where the Patterns for e-business deliver a significant part of the overall solution. In these chapters, the use of the Patterns for e-business will be illustrated within the overall context of the IBM Global Services Method.

2.3.1 Key terms and concepts

Refer to “Glossary” on page 203 for information about key terms and concepts.

2.4 Questions

Answer the following questions. To check your answers, see “Chapter 2: Answers to questions” on page 176.

2.4.1 Discussion and review questions

1. What is a method family? How do engagement models relate to method families?
2. What is the relationship between work products and deliverables?

3. Provide a brief description of the events that occur in the phases of the e-business Custom Engagement Model process provided in Figure 2-3.

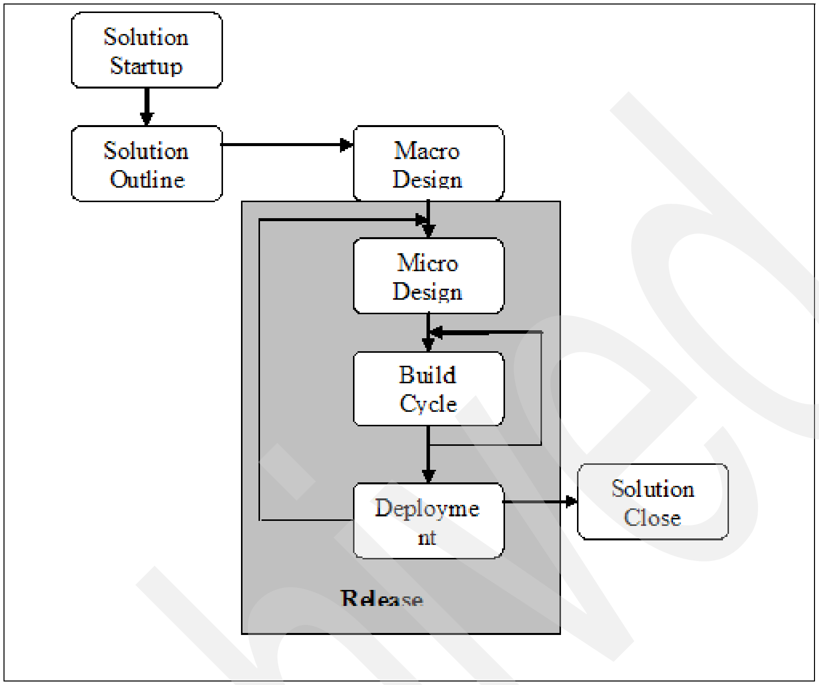


Figure 2-3 Custom Engagement Model

4. What is a Work Product Description (WPD)? How is a WPD organized?

2.5 Exercise and challenge questions

Complete the exercise and challenge questions for further understanding.

2.5.1 Challenge question

Research IBM Global Services' goal of shifting its business model. How does the IBM Global Services Method tie into this organizational goal?

2.5.2 Team project: Walkthrough and Learn Case Study

The three-fold objective of the Walkthrough and Learn Case Study is as follows:

- To illustrate the application of Patterns for e-business best practices resources at various stages of the application/solution development cycle.
- To illustrate how a complex e-business problem can be decomposed into simpler problems, each matched with one of the four primary Business patterns.
- To provide a context for the team to collaboratively walk through a case study for discussion and deeper understanding.

The Walk Through and Learn Case Study is about FutureStep Electronics. The case can be found in pages 18 - 23, 50 - 51; 61 - 63 in *Patterns for e-business: A Strategy for Reuse, by* Adams, et al. We highly recommend that the team read through chapters 2 through 5 (pages 13 - 63) of that book for a better understanding of the case and the application of the method.

FutureStep Electronics, an electronic parts manufacturing company, wishes to use e-business technologies to implement a customer self-service solution that will give its customers comprehensive, around-the-clock service. The solution will provide the following:

- A personalized customer service experience
- Access to electronic component information
- Opinions and buying advice that enable customers to make improved buying decisions
- An easy-to-use component ordering process
- Access to financing from outside financial institutions.

FutureStep would also like to provide access to these business functions across several channels, including Web browsers across the internet, personal digital assistants (PDAs), and pagers.

Team assignment

After reading pages 13 - 63 of *Patterns for e-business: A Strategy for Reuse,* collaboratively walk through the following steps and then discuss them to gain insights and a better understanding of the method and the application of the patterns concept.

- Step 1: Develop a high-level business description.
- Step 2: Develop a Solution overview Diagram.
- Step 3: Identify Business Patterns in the FutureStep Electronics solution.
- Step 4: Identify Integration Patterns in the FutureStep Electronics solution.

2.5.3 Multiple choice

1. The _________________ provides a comprehensive body of knowledge allowing practitioners to conduct engagements in a standardized fashion by relying on predefined engagement models and delivering the overall business solution through the creation of a number of well defined work products.

   a. IBM General Services Model.

   b. IBM General Services Method.

   c. IBM Global Services Model.

   d. IBM Global Services Method.

2. Engagement models describe the typical approaches in delivering a project in terms of:

   a. The work breakdown structure to be adhered to.

   b. Work products to be created.

   c. The roles to be allocated in the project.

   d. All of the above.

3. The _____ describe(s) the sequence of phases, activities, and tasks to be performed in order to deliver the business solution.

   a. Work breakdown structure.

   b. Work products.

   c. Roles to be allocated in the project.

   d. None of the above.

4. _____ refers to the tangible assets produced during an engagement.

   a. Work breakdown structure.

   b. Work products.

   c. Roles to be allocated in the project.

   d. None of the above.

5. During _____, the actual phases, activities, and tasks will be defined, as well as the relevant work products and techniques.

   a. Method definition.

   b. Definition tailoring.

   c. Method tailoring.

   d. None of the above.

6. All _____ are created from _____, but not all _____ are ______.

   a. Deliverables, work products, work products, deliverables.

   b. Deliverables, work products, deliverables, work products.

   c. Work products, deliverables, deliverables, work products.

   d. Trick question, no such thing as work products.

7. A Work Product Description (WPD) describes a particular type of work product and gives guidance on:

   a. How to create the actual work product.

   b. Notation.

   c. The recommended structure of the work product itself.

   d. All of the above.

8. The Work Product Description _____ domain includes WPDs used to define, plan, manage, and close an IBM Global Services Method engagement.

   a. Engagement.

   b. Business.

   c. Organization.

   d. None of the above.

9. The Work Product Description _____ domain provides the WPDs needed to conduct a structured investigation of the current and desired situations in a customer's business in order to develop and implement the appropriate solution.

   a. Engagement.

   b. Business.

   c. Organization.

   d. None of the above.

10. The Work Product Description _____ domain contains the WPDs used to assess organizational and cultural impacts and issues, in order to determine the customer's ability to absorb the change required to develop and implement a new solution.

    a. Operations.

    b. Architecture.

    c. Organization.

    d. None of the above.

11. The Work Product Description _____ domain provides WPDs for the full life cycle of application development, including design, development, testing, and maintenance of applications.

    a. Application.

    b. Architecture.

    c. Organization.

    d. None of the above.

12. The Work Product Description _____ domain contains the WPDs used to structure a solution architecture, including the hardware and software components, the externally visible interfaces, and the relationships among them.

    a. Application.

    b. Architecture.

    c. Operations.

    d. None of the above.

13. The Work Product Description _____ domain includes the WPDs pertaining to the ongoing operations of a solution, including facilities management, problem reporting, information technology management, and security considerations.

    a. Application.

    b. Architecture.

    c. Operations.

    d. None of the above.

14. The domain structure and Work Product Descriptions of the Method can:

    a. Ensure inadequate coverage of the problem domain.

    b. Help organize the definition of key aspects of the solution.

    c. Both A and B are correct.

    d. Neither A or B is correct.

15. The e-business method family engagement model of _____ is intended for medium to large custom, media-rich, interactive application development projects with a major online or new media delivery aspect.

    a. e-business Custom.

    b. e-business Design.

    c. e-business Express.

    d. None of the above.

16. In the e-business Custom engagement model, the majority of new source code produced is:

    a. Media platform based, such as Shockwave, Flash, HTML, and so on.

    b. Object-oriented, such as Java, C++, Smalltalk, and so on.

    c. Both A and B are correct.

    d. Neither A or B is correct.

17. In the _____ phase of the e-business Custom Engagement Model process, the focus of the activities is to tailor the engagement model to determine the actual phase/activity/task structure that will be followed during the project and work products that will be created throughout the project.

    a. Solution Startup.

    b. Solution Outline.

    c. Macro Design.

    d. None of the above.

18. During the _____ phase of the e-business Custom Engagement Model process, the high level business requirements, business objectives, and IT objectives are taken into consideration, focusing each time on those requirements/objectives influencing the overall architecture.

    a. Solution Startup.

    b. Solution Outline.

    c. Macro Design.

    d. None of the above.

19. During the Solution Outline phase of the e-business Custom Engagement Model process, the area of _____ within the requirements domain is considered.

    a. Current IT environment.

    b. Security.

    c. Business Impact.

    d. All of the above.

20. During the _____ phase of the e-business Custom Engagement Model process, the overall architecture of the IT system is specified, which allows one to define solution-wide approaches for addressing both functional and non-functional requirements.

    a. Macro Design.

    b. Micro Design.

    c. Deployment.

    d. None of the above.

21. During the Macro Design phase of the e-business Custom Engagement Model process:

    a. The specification of the required training and user support is developed.

    b. Refinement of the architecture based on the technology and product selection occurs.

    c. Both A and B are correct.

    d. Neither A or B is correct.

22. The purpose of the _____ phase of the e-business Custom Engagement Model process is to incrementally develop and test part of the system until the objectives of the overall release are achieved.

    a. Micro Design.

    b. Build Cycle.

    c. Deployment.

    d. None of the above.

23. During the _____ phase of the e-business Custom Engagement Model process, the results of the project are assembled, analyzed, and archived for use by other practitioners.

    a. Build Cycle.

    b. Deployment.

    c. Solution Close.

    d. None of the above.

2.5.4 Fill in the blank

1. Method families reflect the close relationships between a number of ____________________ applicable to the same domain.

2. During ______________________, the actual phases, activities, and tasks will be defined, as well as the relevant work products and techniques.

3. Work Product Descriptions cover a broad range of work performed on engagements and have been further organized into __________________.
4. The Work Product Description _______________ domain contains the WPDs used to assess organizational and cultural impacts and issues to determine the customer's ability to absorb the change required to develop and implement a new solution.
5. ____________________ models provide guidance on the required phases, activities, and tasks, the Work Product to be produced, the roles assigned to the project, and any applicable techniques.
6. A Work Breakdown Structure decomposes what must be done to fulfill objectives into progressively ________________ units of work.
7. _________________ are defined as increments of the overall solution that are effectively made available.
8. The purpose of the ______________________ phase is to deploy the system and prepare for the next release of the system.
9. The ______________________ phase and _____________________ phase are the bookends to every engagement.

2.5.5 True or false

1. The IBM Global Services Method is organized around a number of method families.
2. The need for method tailoring underlies the fact that the IBM Global Services Method is unable to cater to many different types of engagements.
3. The Work Product Description architecture domain provides WPDs for the full life cycle of application development, including design, development, testing, and maintenance of applications.
4. A Work Breakdown Structure (WBS) breaks down the work of the project into a hierarchical structure that includes the phases of the engagement model as well as activities and tasks.
5. The Work Breakdown Schedule can be transformed into a project plan by adding the dimensions of time and resources.
6. A key benefit of working during proposal development with a subset of work products relevant to the solution delivery project is that the proposed solution is already expressed in terms of work products that will be used by the delivery team.
7. All key stakeholders must clearly understand and support the scope and objectives of project.

8. The purpose of the Micro Design phase is to prepare for the Build Cycle of a specific release of the system by driving the architecture and design to a release and implementation specific view.
9. Within a specific release, multiple Build Cycles could be planned, each with their own objectives in line with and contributing to the overall Release specific objectives.
10. The primary objective of the Solution Startup phase is to determine the complexity and scope of the solution.

2.5.6 Short answer

1. What is a method family? How do engagement models relate to method families?
2. What is a Work Product Description (WPD)? How is a WPD organized?
3. Provide a brief description of the events that occur in the phases of the e-business Custom Engagement Model process provided in Figure 2-4.

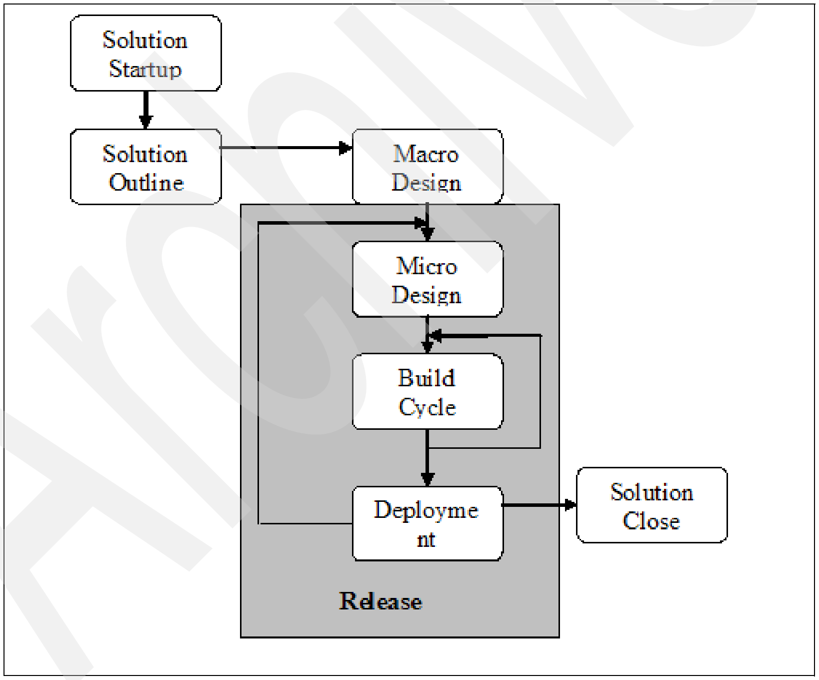


Figure 2-4 Custom Engagement Model

3

Building solutions on a Single Business pattern basis

In this chapter, we explore how the Patterns for e-business can contribute to the development of a solution based on a single Business pattern. It will be made clear that the Patterns for e-business are not a replacement for any development method in use during solution design or delivery, but are essentially an asset selection technique together with a rich set of associated assets that can be used during the phases/activities of a solution design/delivery project. In order to relate the contribution of the Patterns for e-business to the various phases and activities of solution design (project proposal) and solution delivery, the overall discussion has been framed within a high level description of the overall phase/activity structure of the e-business Custom engagement model of the IBM Global Services Method.

Outline

- Using the Patterns for e-business during proposal development
  - Case study for proposal development
    - Identifying requirements
    - Documenting the client environment
    - Defining the solution
    - Summary
- Using the Patterns for e-business during proposal development
  - Solution Startup phase
  - Solution Outline phase
  - Macro Design phase
  - Micro Design phase
  - Build Cycle phase
  - Deployment phase
- Case study for solution delivery
  - Solution Startup phase
    - Defining the project approach
- Solution Outline phase
  - Identifying requirements
  - Defining the high lever architecture
    - Selecting the Business pattern
    - Selecting the Application pattern
    - Selecting the Runtime pattern
    - Selecting the Product mapping
- Macro Design phase
  - Identifying requirements
  - Defining the architecture
  - Building the development environment
- Micro Design phase
  - Refining the architecture
- Build Cycle phase
  - Building the production environment

- Deployment phase
  - Reviewing the solution plans
- Summary

Learning objectives

The objectives of this chapter help you to:

- Understand and explain a solution built on the Single Business Pattern.
- Understand and explain the value added by IBM Patterns for e-business.
- Explain the reasoning behind the chosen solution.
- Explain the part of the chosen solution and how these parts interact.

Real-world environment

In this section, we refer you to a real-world situation that uses the concepts in this chapter. The case study, called “Degussa-Huls divisions go to market with IBM e-business solutions”, is located on the Web at:

`http://www.redbooks.ibm.com/additional_materials/SG247466/Degussa-Huls_Divisions.pdf`

3.1 Using the Patterns for e-business during proposal development

Table 3-1 gives an overview of some of the activities occurring during the development of a project proposal. Typically, these activities can take place in various contexts, such as during the preparation of a proposal linked to a particular Request for Proposal (RFP) or during the definition of a project proposal by an internal customer team. We will focus on those activities for which the assets of the Patterns for e-business are of most value.

Table 3-1 Proposal development and the IBM Patterns for e-business

Proposal Development (SolutionDesign) activities	**Added value of the IBM Patterns for e-business**
Identifying Requirements	The navigation of the Patterns for e-business asset catalog starts from a broad categorization of business requirements. Not only do these guide the asset selection process but they also offer additional insights into the overall business area and the main business drivers.
Documenting the Client Environment	The information found in the request for proposal (RFP) or available to the customer team will help to document the current IT environment of the client. This information is key during the later stages of the asset selection process and at the same time creates an opportunity for a focused analysis of the request for proposal or the available documentation.
Defining the solution	Patterns for e-business contribute significantly to the definition of the solution to be proposed.

3.1.1 Case study for proposal development

We will now discuss each of the activities outlined in Table 3-1 in the context of the First Financial case study.

Identifying requirements

Based on requirements communicated by First Financial, a large financial services company, the project proposal team has developed the following understanding of the business context.

First Financial is looking to provide a Web channel for their existing credit card system. To accomplish this, they wish to Web-enable the existing CICS credit card application for account access, rewards programs, credit line increases, and so on.

First Financial's business objectives are:

- Reduction of time to market
- Tactical focus on only supporting the Web channel
- Strategic focus on supporting multiple access channels, such as call center, telephone access, and so on
- Leverage existing investments and the existing CICS skill base

Documenting the client environment

The key elements characterizing the customer's environment are:

- A requirement to access an existing CICS-based application
- A strong emphasis on the privacy of the exchanges between the front-end and the existing applications and the authentication of a client before any access to an existing application is allowed
- A preference for Windows® 2000/System x™ servers to host the Web infrastructure
- A current installed base of either Windows 2000/System x or System z

Defining the solution

Based on the information gathered from First Financial (and summarized in the previous sections), we decide to use the Patterns for e-business for a fast path definition of a suitable and proven solution meeting the business requirements of the company. This fast path approach is only expected to provide an 80% fit for the full set of First Financial's requirements, so a comprehensive development method will still be needed to customize these assets to result in a 100% fit.

These patterns should be considered candidate assets that leverage the experience of architects to create solutions quickly, whether for small local businesses or large multinational enterprises. Specifically in the context of the development of a project proposal, the ability to quickly translate requirements into a coherent and viable solution is extremely valuable. Also, the proven product configurations documented in the Patterns for e-business asset catalog contribute significantly to producing a high quality project proposal on time.

The key elements contributed by the Patterns for e-business are:

- Identification of required functionalities
- Product selection guidance
- Information regarding the interactions between products

For the purposes of the proposal, the following four-step procedure is used to arrive at the recommended solution:

1. Selection of the Business pattern(s):
   - An analysis of the high-level business requirements described above indicates that the Self-Service Business pattern may be applicable to this solution.
   - Self-Service Business pattern

     In this particular case, First Financial is interested in extending an existing credit card application through a Web channel to its customer/user base. The Self-Service Business pattern is applicable if the requirements suggest that users need to interact directly with existing data.
2. Selection of the Application pattern(s):

   The Application pattern focuses on the application level functionality required to support the identified Business pattern(s).

   Based on the available information, the Router Application pattern is selected.

   - Router Application pattern

     First Financial plans to provide access to its existing back-end systems through different channels (initially only a Web channel, but the call center and telephone have been identified as future extensions). The Router application pattern provides a structure for applications that require the intelligent routing of requests from multiple delivery channels to one of multiple back-end applications.
3. Selection of the Runtime pattern(s):

   The Runtime pattern focuses on the technical functionality needed to support an Application pattern.

   - Router Runtime pattern

     In this Runtime pattern, the functions of the router tier are performed by an integration server.

4. Obtaining the Product mapping:

   Given the existing base of Windows 2000/System x servers, the Product mapping based on a deployment on Windows 2000/System x servers is selected.

   This Product mapping gives us immediately an overview of the required products and their relationships.

   Based on the information obtained from the Patterns for e-business, the proposal can then be further developed by:

   – Estimating the man days needed to build new functionality or to integrate existing functionality
   – Estimating the software budget, taking into account, among other elements, the suggested software products
   – Specifying the resource profiles required to deliver the solution (among others derived from selected products)
   – Specifying the tooling required for development (among others derived from selected products)

Note: Other factors are crucial in being able to realistically budget for the proposed solution, like sizing estimates, available skills and other relevant non-functional requirements (availability, performance, and so on), but these are outside the scope of this discussion.

3.1.2 Summary

This section shows how the Patterns for e-business can be used very effectively in defining (a part of) the solution in response to a request for proposal. The Patterns for e-business allow us to arrive very quickly at such a solution through an asset selection process starting from an analysis of the main business and IT drivers present in the request for proposal. The obtained results, expressed in terms of required software products, can be used immediately to drive the budgeting aspect of the solution design and will increase the overall technical quality of the proposal given its reliance on the proven software and hardware product combinations found in the Patterns for e-business asset catalog.

3.2 Using the Patterns for e-business during solution delivery

In Table 3-2, we give an overview of the various activities undertaken during solution delivery. We will focus on the activities related to requirements gathering and architecture definition because the Patterns for e-business provide the most added value in these areas by:

- Clarifying the applicable requirements (which are subsequently used to drive the selection of the applicable patterns and assets)
- Delivering standard application and product configurations (which can reduce the architecture related development effort)

Table 3-2 IBM Global Services Method and IBM Patterns for e-business

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Phase: Solution Startup	
Defining the project approach	Starting from the approach outlined in the project proposal, confirm and refine the proposed approach based on any additional information. This is usually done both with the project team and client personnel. The use of the IBM Patterns for e-business will be taken into account during this stage since this will help to shape the overall approach.
Phase: Solution Outline	
Identlfying requirements	The focus of this activity is to gather the overall business requirements and objectives, concentrating on those that will have a significant impact on the solution. One technique that supports getting an overall view of the solution's requirements involves the development of a system context diagram. This diagram represents the proposed solution, emphasizing the interactions to be supported with its environment, both in terms of people or systems. The Business patterns can provide richness to this diagram by relating it to similar trends in other industries while at the same time allowing for the validation of the selected pattern(s).

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Defining the high level architecture	The main objective of this activity is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be given to the more challenging areas of the proposed solution. Based on the architecturally significant requirements, the architect will validate the business and IT drivers from the client and match these against the ones associated with the patterns. This process will then lead to the identification of the involved Application patterns. The applicable Application patterns can be documented as part of an architecture overview diagram that would feature, for example, the Router Application pattern. An adopted Application pattern will lead the architect to a specific set of Runtime patterns and Product mappings, taking into account the actual IT environment of the customer. It is very important that at this stage an accurate assessment be made of any gaps between the identified assets and the actually required solution so that adequate attention can be given to those areas through either the selection of additional products or through custom development.
Phase: Macro Design	
Identifying requirements	The focus of this activity is to gather and refine all applicable requirements (instead of focusing only on the architecturally significant ones) so that the results from the Identifying requirements activity performed during the Solution Outline phase can be confirmed and further refined.

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Defining the architecture	This activity will focus on specifying the overall architecture, in terms of how it addresses the functional and non-functional requirements so that in subsequent phases, a start can be made on its implementation. During this activity, the architecture defined during the Solution Outline phase will be confirmed and further refined. This time we will review the Patterns selected during the previous phase based on all known requirements. Once the Business patterns have been confirmed, the following steps can be taken: 1. Finalize the Application patterns to be used. 2. Select the Runtime pattern for each of the identified Application patterns. 3. After selecting the Runtime patterns, an experienced architect should integrate these patterns into the overall architecture. 4. Perform the Product mapping on the basis of the current IT environment of the customer. For the selected Runtime patterns, review the Product mappings on the Patterns Web site at: `http://www-106.ibm.com/developerworks/patterns/u2b/at5-product-map.html` This information can be used as a guideline for selecting the products in the case where they have not yet been defined. In situations where the client has already predefined an infrastructure, the information on this Web site can be used to validate the decisions that have been made, again based on the previously identified Business, Integration, Composite, Application and Runtime patterns.
Building the development environment	Review the application development chapters of the applicable pattern IBM Redbooks publications to support the definition of the development environment. Additionally, review product installation instructions in the applicable pattern IBM Redbooks publications to assist with the product installs for the development environment.

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Phase: Micro Design	
Refining the architecture	The main objective of the Micro Design phase is to complete the design so that the build activities can be performed. Guidance on recommended design practices can be found in the application design and application development chapters of the applicable pattern IBM Redbooks publications.
Phase: Build Cycle	
Building the production environment	The focus of this activity is to build the production environment based on the architecture defined during the previous phases. Again, refer to the product installation instructions in the applicable pattern IBM Redbooks publications to assist in building the production environment. This may be especially helpful if there are differences between the development/test and production environments.
Phase: Deploy	
Reviewing the solution plans	Review the system management guidelines chapters in the applicable pattern IBM Redbooks publications.

From this overview, it should be clear that the Patterns for e-business are not a replacement for any development method in use during solution design or delivery, but are essentially an asset selection technique together with a rich set of associated assets that can be used during the phases/activities of a solution design/delivery project. In order to relate the contribution of the Patterns for e-business to the various phases and activities of solution design and solution delivery, the overall discussion has been framed within a high level description of the overall phase/activity structure of the e-business Custom engagement model of the IBM Global Services Method. It should be clear that the Patterns for e-business need to be included in a comprehensive project development approach in order to be able to deliver 100% of the business requirements.

3.3 Case study for solution delivery

Now we begin the Solution Delivery for First Financial.

3.3.1 Solution Startup phase

During this phase, we seek to confirm the overall approach for the project as originally defined in the proposal. An important input to this activity is the information previously made available by First Financial and the actual solution proposed. Particularly relevant for this phase is the information related to the possible use of assets during the engagement.

For the purposes of this section, we will assume that the problem setting introduced in the proposal development section is still valid.

Defining the project approach

The focus of this activity is to familiarize the project team with the development method, assessing training and mentoring needs and tailoring the development method to the needs of the project by taking the use of assets into account.

An important step in starting any project is to become familiar with any candidate assets that may be useful during the development of the solution. The use of assets can improve the profitability of a project as well as lower its risk given that the solution will be based on assets used in previous successful implementations. The candidate assets we are particularly interested in are the Patterns for e-business. The starting point for the work undertaken on First Financial's project is the architecture defined in the project proposal.

3.3.2 Solution Outline phase

During the Solution Outline phase, the overall shape and style of the system is decided based upon an analysis of the key requirements. The emphasis during this phase lies much more on identifying all the required functionalities than on exploring each one in detail. The overarching objective is to get an accurate view of the overall effort required to deliver the system. Consequently, attention will be paid specifically to those areas deemed to be particularly challenging.

Identifying requirements

The objective of the “Identifying requirements” activity during the Solution Outline phase is to gather the overall business requirements and objectives, particularly concentrating on those which will have a significant impact on the solution.

Starting with the requirements expressed by First Financial and the way these were interpreted in the project proposal, we will reexamine the customer's requirements in more detail.

If this initial information is not available, then some of the steps normally taken during the proposal development will be performed at this time, for example, identifying the high level requirements.

During this phase of the project, we want to obtain more detailed requirements so that we can identify the actors and key business functions to be delivered by the proposed solution.

During the requirements gathering effort, involving First Financial's personnel, the following additional non-functional requirements were obtained:

- The application must support multiple languages.
- Performance must be enhanced over existing Web-based applications.
- Capacity/logon (500 logons and fifty transactions every minute).
- Scalability (10,000 users in two years).
- Security (secure transaction support; strong encryption and connections to CICS are only allowed if authenticated).
- Technology constraints (front-end access is Netscape and IE browser-based, CICS V4.1 for existing setups).

Based on all available requirements, we develop a system context diagram illustrating which interactions with users and other systems will need to be supported by the new system. This diagram also confirms our use of the Self-Service Business pattern as it supports direct interactions between interested parties and a business.

As you use the patterns, it is important to keep in mind that, although we may frequently refer to the “users” of a system, in many cases the user can be almost any person, process, or thing, provided the pattern fits the situation and the user exercises the business process.

The system context diagram for First Financial is shown in Figure 3-1.

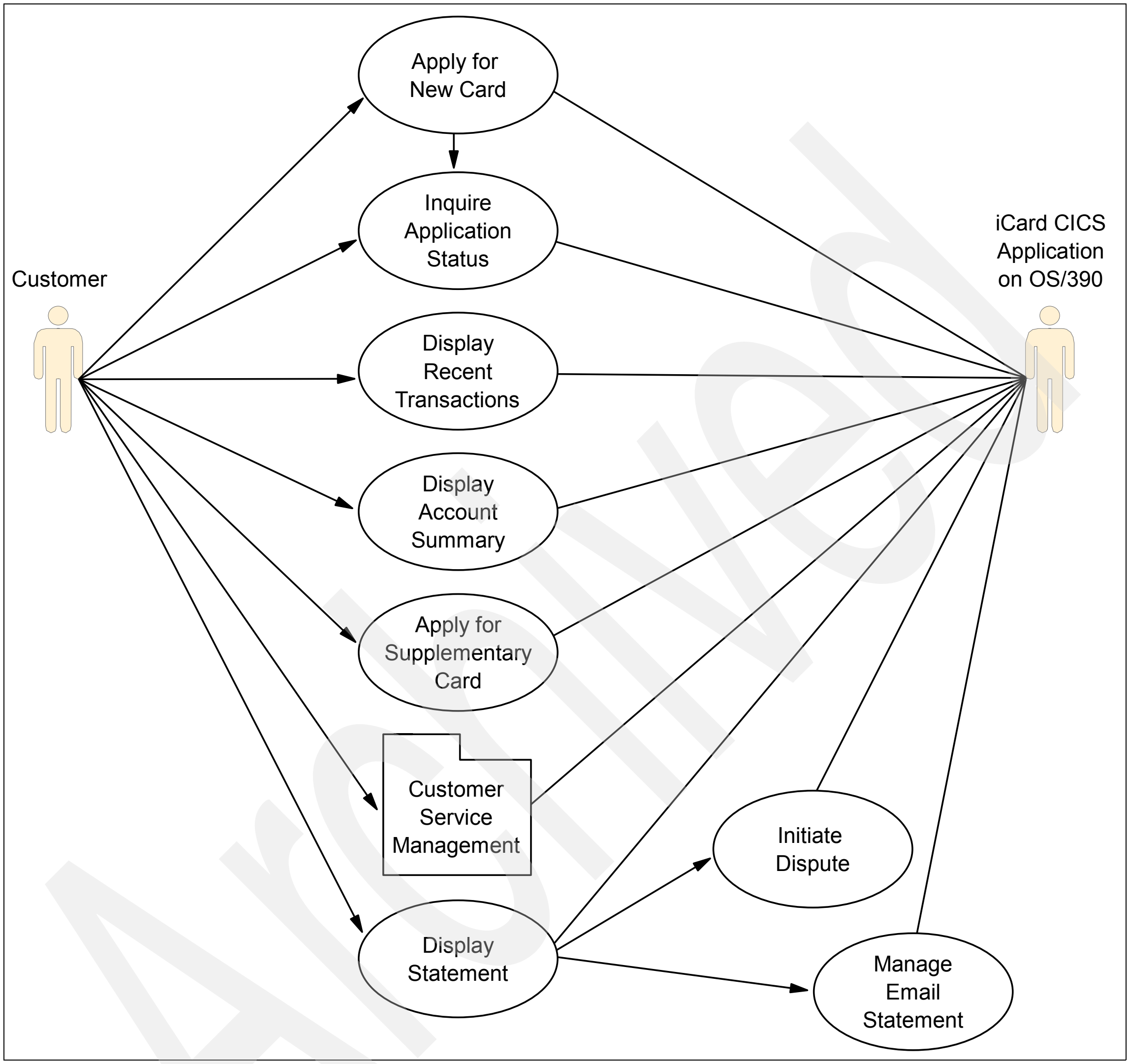

Customer
Apply for New Card
Inquire Application Status
Display Recent Transactions
Display Account Summary
Apply for Supplementary Card
Customer Service Management
Display Statement
Initiate Dispute
Manage Email Statement
iCard CICS Application on OS/390

Figure 3-1 Scenario - First Financial

The Patterns for e-business can also be useful for identifying requirements by drawing attention to various business and IT drivers that will be used as input to the Application pattern selection process. At the same time, focusing on these key requirements (or key drivers) will facilitate the subsequent identification of the relevant patterns.

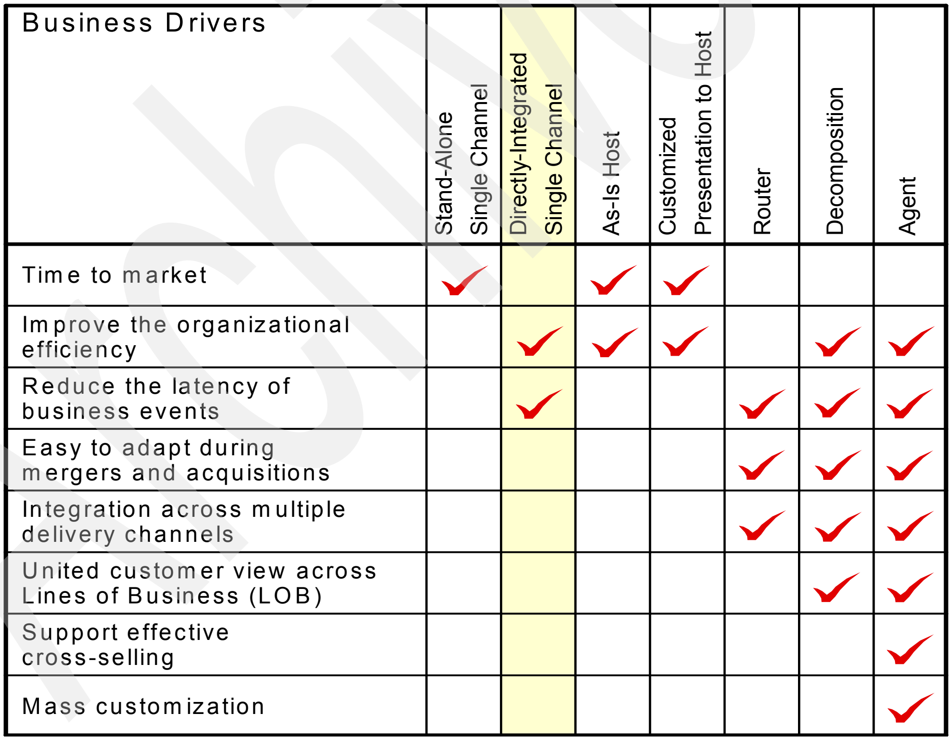
Defining the high level architecture

The main objective of this activity is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be paid to the more challenging areas of the proposed solution.

In this particular case, we will focus on the use made of the Patterns for e-business assets. Also, we want to reevaluate the architectural decisions taken during proposal development on the basis of all requirements known at this stage.

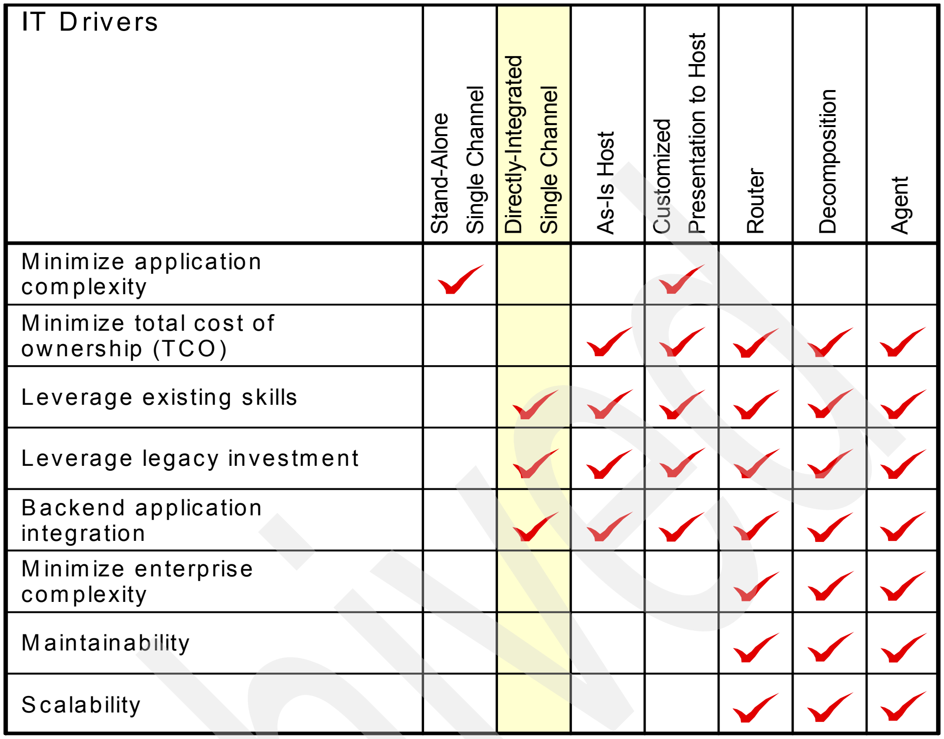
On the basis of the currently known requirements, we confirm the Self-Service Business pattern as the Business pattern most closely supporting our requirements.

Figure 3-2 and Figure 3-3 on page 64 present the typical business and IT drivers (shown as row headers) versus the Application patterns (shown as column headers) for the Self-Service Business pattern.



Business Drivers	Stand-Alone Single Channel	Directly-Integrated Single Channel	As-Is Host	Customized Presentation to Host	Router	Decomposition	Agent
Time to market	✓		✓	✓			
Improve the organizational efficiency		✓	✓	✓		✓	✓
Reduce the latency of business events		✓			✓	✓	✓
Easy to adapt during mergers and acquisitions					✓	✓	✓
Integration across multiple delivery channels					✓	✓	✓
United customer view across Lines of Business (LOB)						✓	✓
Support effective cross-selling							✓
Mass customization							✓

Figure 3-2 Business Drivers of the Self-Service Business pattern



IT Drivers	Stand-Alone Single Channel	Directly-Integrated Single Channel	As-Is Host	Customized Presentation to Host	Router	Decomposition	Agent
Minimize application complexity	✓			✓			
Minimize total cost of ownership (TCO)			✓	✓	✓	✓	✓
Leverage existing skills		✓	✓	✓	✓	✓	✓
Leverage legacy investment		✓	✓	✓	✓	✓	✓
Backend application integration		✓	✓	✓	✓	✓	✓
Minimize enterprise complexity					✓	✓	✓
Maintainability					✓	✓	✓
Scalability					✓	✓	✓

Figure 3-3 IT Drivers of the Self-Service Business pattern

The next step is to select an appropriate Application pattern. A salient feature of the patterns is that although each of us may work with customers in many different industries, there are a number of well understood, common approaches that work well across industries. Also remember that although patterns are ways of associating requirements and solutions, there are always different ways of physically implementing them.

In reviewing the Business and IT drivers from Figure 3-2 on page 63 and Figure 3-3 above, we can confirm that the Self Service::Router Application pattern is still the best fit. The Multiple Channel requirement maps clearly back to the key Business driver expressed by First Financial and Leveraging Legacy Systems is one of their main IT related concerns. These two support the use of the Router Application pattern. Further support for using this pattern is as follows.

The Router Application pattern supports multiple delivery channels connecting to one or more back-end existing applications through a router tier. This router tier can route a single message to one or more applications within the same business process. This is in line with First Financial's requirements, as noted in

Figure 3-4. The router tier can process a single request within this business line and access the required existing systems to produce a response, in this case, the rating and underwriting systems. Refer to Figure 3-5 on page 66 for an example of the Router Application pattern.

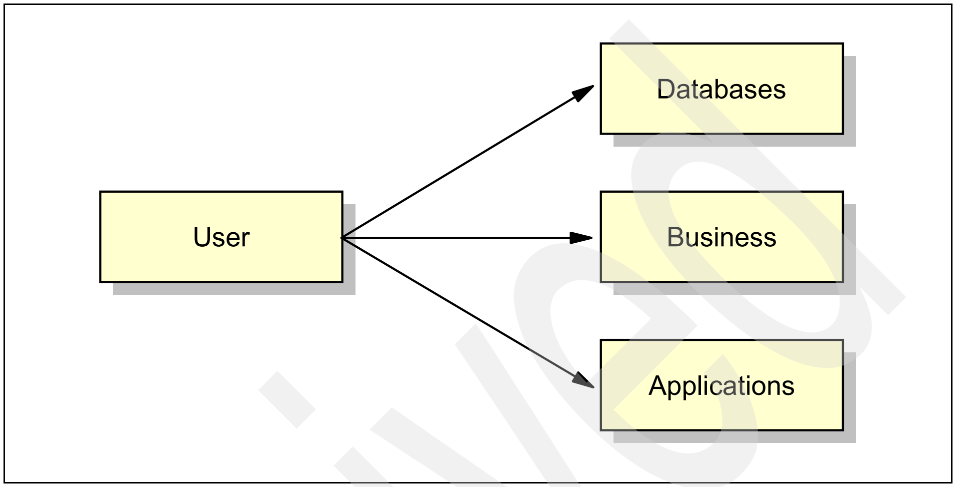


Figure 3-4 Self Service, User to Multiple Applications

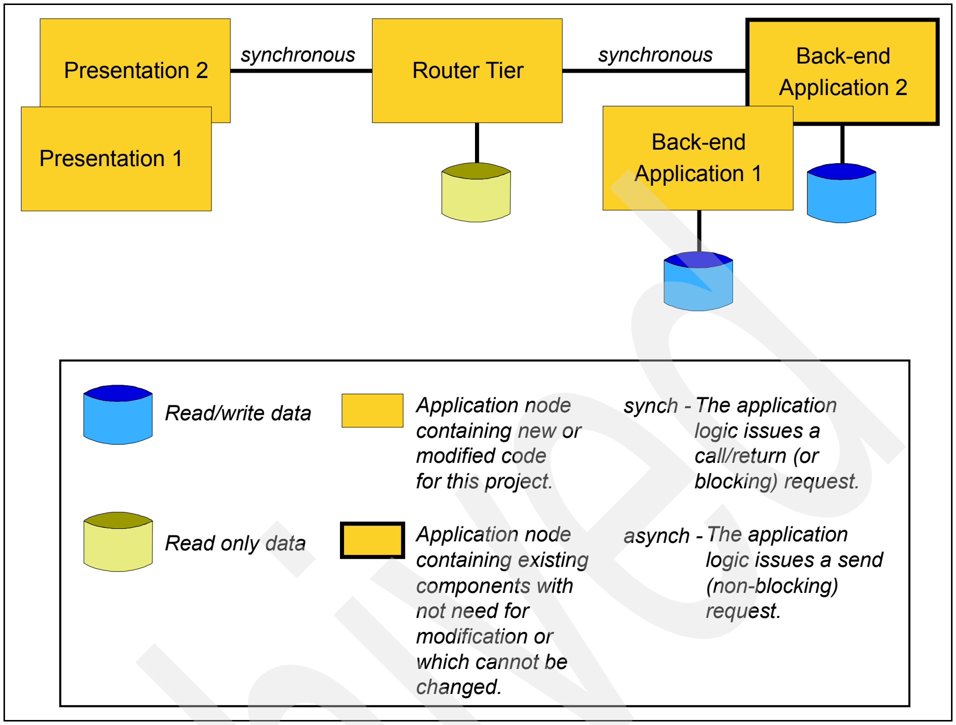

Presentation 2
Presentation 1
synchronous
Router Tier
synchronous
Back-end Application 2
Back-end Application 1
Read/write data
Read only data
Application node containing new or modified code for this project.
Application node containing existing components with not need for modification or which cannot be changed.
synch - The application logic issues a call/return (or blocking) request.
asynch - The application logic issues a send (non-blocking) request.

Figure 3-5 Self Service::Router Application pattern

The overall solution can be documented through an architecture overview diagram showing how the Router Application pattern integrates within the overall proposed solution.

The selection of an Application pattern leads to a set of specific Runtime patterns.

In the Router Application pattern, the router tier serves as an integration point for delivery channels in the presentation tier, allowing access to individual back-end applications. In the following Runtime pattern, which supports the Router Application pattern, the functions of the router tier are performed by an integration server. Figure 3-6 on page 67 shows a Runtime pattern for Self-Service::Router.

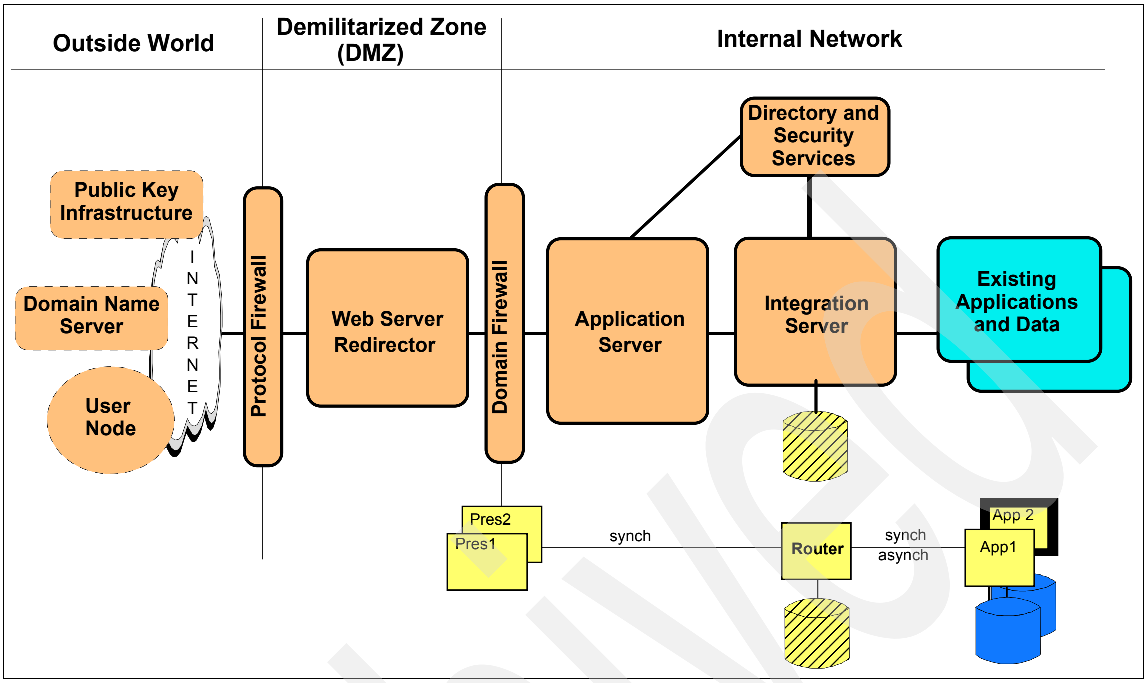


Figure 3-6 Runtime pattern- Self-Service::Router

Each of the nodes identified in Figure 3-6 is described in detail on the Patterns for e-business Web site. Refer to the following URL for more information:

http://www-106.ibm.com/developerworks/patterns/u2b/at5-runtime.html

Once the Runtime pattern has been established, the corresponding Product mapping can be defined, taking into account the actual IT environment of First Financial. At this stage, we will only concentrate on having an overview of the products that will support the overall solution. This is done with a view to make an accurate assessment of any gaps between the identified assets and what is needed to construct the solution. In this way, adequate attention can be given to the definition of those areas for which additional products or custom development will be required. Also, we will focus on the integration points between the involved products in order to asses the effort involved in effectively achieving their integration. Given that the product configurations proposed by the Patterns for e-business reflect proven solutions, this integration will not pose significant problems.

3.3.3 Macro Design phase

During the Macro Design phase, we begin to develop the complete specification of the architecture in line with the work carried out during the Solution Outline phase. The Macro Design phase also lays the groundwork in preparing for the subsequent releases by developing a robust user interface and architecture capable of withstanding reasonable business change through the delivery of these releases and beyond. The Patterns for e-business will further support the activities undertaken at this stage.

Identifying requirements

Whereas “Identifying requirements” during the Solution Outline phase referred to the activity of determining the key requirements influencing the overall architecture, we will now concentrate on refining the system specification based on all user requirements, business processes, and business requirements by elaborating the system requirements and specifications. This will enable us to:

- Make better estimates for the first release.
- Provide an overall specification for the structure and topology of the system.

Defining the architecture

Many of the activities in this phase are similar to those carried out during the Solution Outline phase. In this phase, we will focus on specifying the overall architecture, both from a structural and topological point of view so that in a subsequent phase, a start can be made on its implementation. During this activity, confirmation will be sought of the decisions taken during the Solution Outline phase.

This is an opportunity to update the architecture overview diagrams based on any new information obtained thus far.

This effectively implies that we will verify during the Macro Design stage that any additional functionalities are catered for by the products suggested by the Patterns for e-business catalog or that these functionalities will be delivered by other products or earmarked for custom development.

During this stage, the Product mappings will be refined to include the exact specification of product levels to be used in putting the solution together. A suggested Product mapping for First Financial's project is shown in Figure 3-7 on page 69.

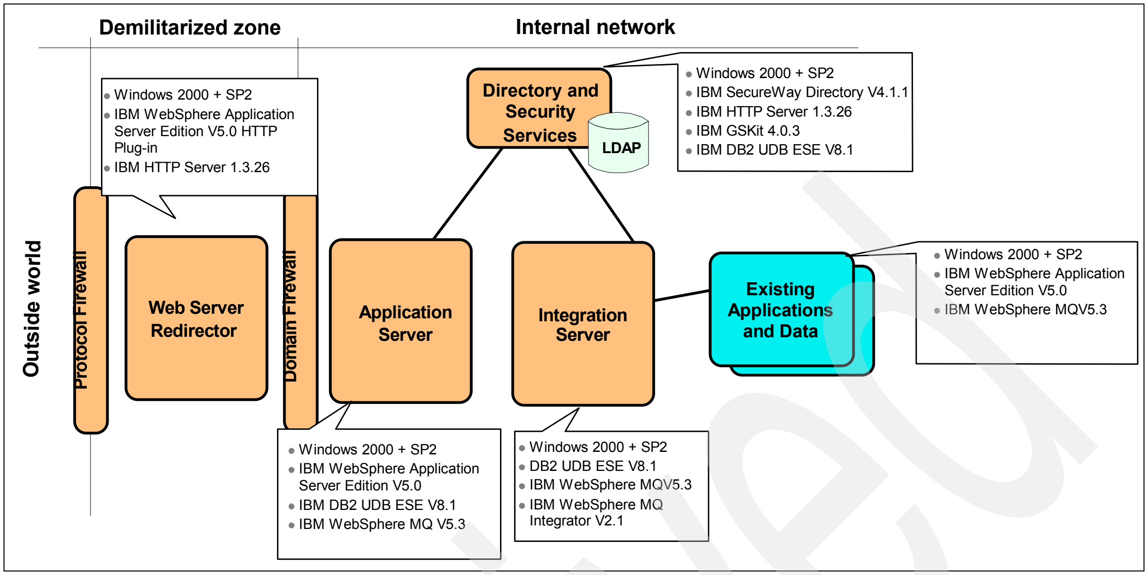


Figure 3-7 Product mappings for Self-Service::Router

> **Note:** The source for all Product mapping information can be found on the Patterns for e-business Web site at the following URL:
>
> `http://www-106.ibm.com/developerworks/patterns/select-pattern.html`

Building the development environment

The purpose of 'Building the development environment” is to put the hardware and software into place to support the development activities and any associated tools.

The Patterns for e-business IBM Redbooks publications can also be leveraged in building this development environment. For each of the involved products, identified through the Product mappings, the architect should refer to the product installation instruction chapters in the related Patterns for e-business series IBM Redbooks publications.

3.3.4 Micro Design phase

The purpose of the Micro Design phase is to prepare for the build cycle(s) of a specific release of the system. This implies a further detailing of the results obtained in the prior two phases but narrowed down to the scope of the release itself. In this way, the foundation is established for all subsequent implementation related activities.

Refining the architecture

The main objective of the “Refining the architecture” activity during the Micro Design phase is to complete the design so that the build activities can be performed.

For design guidance related to a particular product, refer to the design and development chapters in Patterns for e-business series IBM Redbooks publications.

3.3.5 Build Cycle phase

The purpose of a Build Cycle phase is to incrementally develop and test part of the system until the objectives of the overall Release, within which the Build cycle is embedded, are achieved.

Building the production environment

During this activity, we want to take a look again at the system's topology defined during the earlier phases and leverage this information to build the production environment. The Patterns for e-business series IBM Redbooks publications can further support building the production environment by referring to the product installation instruction chapters.

3.3.6 Deployment phase

The purpose of this phase is to deploy the system and prepare for the next release of the system.

Reviewing the solution plans

The purpose of this activity is to complete the systems management and software distribution plans.

This provides the basis for:

- Selecting system management tools.
- Selecting systems management processes.

- Selecting and defining job descriptions, based on identified roles, to support the IT system.

For more information about system management guidelines, refer to the system management processes guideline chapters in Patterns for e-business IBM Redbooks publications.

3.4 Summary

In this chapter, we have shown how to start from a business case and define a solution on the basis of a single Business pattern. We illustrated this for two different points in time, initially during proposal creation and later on during solution delivery. Each time, we demonstrated how the Patterns for e-business can contribute significant value. The assets of the Patterns for e-business can deliver up to 80% of the proposed solution, and lend themselves very well to integration into an overall solution while at the same time making an important contribution to the overall quality of the solution.

3.4.1 Key terms and concepts

Refer to “Glossary” on page 203 for a list of key terms and concepts.

3.5 Questions

Answer the following questions. To check your answers, see “Chapter 3: Answers to questions” on page 181.

3.5.1 Discussion and review questions

1. What value is added by the IBM Patterns for e-business in the various Solution Delivery Phases of the IBM Global Services Method?
2. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Proposal Development activities, based on the case provided, please provide the following:
   a. What is the scope of the intended project based on the results of the Identifying Requirements activity?
   b. What are the Business Objectives relevant to the project?

    c. What are the key elements of the client environment relevant to the project?

3. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Proposal Development activities, based on the case provided, please provide the following:

    a. Which Business Pattern was chosen? Why?

    b. Which Application Pattern was chosen? Why?

    c. Which Runtime Pattern was chosen? Why?

4. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Solution Delivery Phase: Solution Outline, based on the case provided, please provide the following:

    a. What is the overall objective of the "Identifying Requirements" activity?

    b. What additional non-functional requirements were identified during this activity?

5. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. Which application pattern was selected? Why?

6. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. How does the Router Application pattern support multiple delivery channels?

7. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. Based on the case provided and with the completion of the Macro Design Phase:

    a. What is the overall objective of the "Identifying Requirements" activity?

    b. How does this differ from the same activity completed during the Solution Outline phase?

3.6 Exercise and challenge questions

Complete the exercise and challenge questions for further understanding

3.6.1 Challenge questions

1. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Proposal Development activities, based on the case provided, please provide the following:

   a. What items are identified in the text as areas that can be further developed?

   b. Conduct research on a financial organization similar to First Financial. Choose one of the items identified above. Expand on one or more of the items identified. What other factors are needed to realistically budget for the proposed solution that are not identified in the text? Make sure you can defend your answer with findings from research.

3.6.2 Team case study projects

Here we discuss the team case study projects.

Proposal development activities

- Case Study 1: Mohawk Industries

  Refer to “Real-world environment” on page 29, which directs you to the case study “Mohawk Industries keeps its customers covered with e-business solutions from IBM.”

  Review 3.1, “Using the Patterns for e-business during proposal development” on page 52, and 3.1.1, “Case study for proposal development” on page 52. Use these sections to guide your development of a project proposal for Mohawk Industries.

  Team Assignment:

  Based on the information communicated in the Mohawk Industries case, collaboratively analyze the following proposal development activities and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach:

  a. Identify the requirements

  b. Document the client environment

  c. Define the solution

- Case Study 2: Degussa-Huls Divisions

  Refer to “Real-world environment” on page 51, which directs you to the case study “Degussa-Huls divisions go to market with IBM e-business solutions.”

Review 3.2, “Using the Patterns for e-business during solution delivery” on page 56, and 3.1.1, “Case study for proposal development” on page 52. Use these sections to guide your development of a project proposal for Mohawk Industries.

Team Assignment:

Based on the information communicated in the Degussa-Huls Divisions case, collaboratively analyze the following proposal development activities and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach:

a. Identify the requirements.

b. Document the client environment.

c. Define the solution.

Development definition of a proposal

- Case Study 3: Mohawk Industries

  Team Assignment:

  From the results of your team's analysis of Mohawk Industries above, apply the following four-step procedure to arrive at the recommended solution and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

  Step 1: Selection of the Business pattern(s)

  Step 2: Selection of the Application pattern(s)

  Step 3: Selection of the Runtime pattern(s)

  Step 4: Obtaining the Product mapping.

- Case Study 4: Degussa-Huls Divisions

  Team Assignment:

  From the results of your team's analysis of Degussa-Huls Divisions above, apply the following four-step procedure to arrive at the recommended solution and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

  Step 1: Selection of the Business pattern(s)

  Step 2: Selection of the Application pattern(s)

  Step 3: Selection of the Runtime pattern(s)

  Step 4: Obtaining the Product mapping.

Solution delivery

- Walkthrough and Learn Case: First Financial

  Team Assignment:

  Collaboratively walk through 3.3, “Case study for solution delivery” on page 60 and analyze the activities involved in the phases shown below and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

  a. Solution Start-up phase

     i. Defining the project approach

  b. Solution Outline phase

     i. Identifying requirements

     ii. Defining the high level architecture

  c. Macro Design phase

     i. Identifying requirements

     ii. Defining the architecture

     iii. Building the development environment

  d. Micro Design phase

     i. Refining the architecture

  e. Build Cycle phase

     i. Building the production environment

  f. Deployment phase

     i. Reviewing the solution plans

- Case Study 5: Mohawk Industries

  Team Assignment:

  From the results of your team's analysis of Mohawk Industries in Case studies 1 and 3, perform the first three phases of solution delivery for Mohawk Industries and then perform a project debriefing to gain insights and a better understanding of the phases and the application of the patterns approach.

  a. Solution Start-up phase

     i. Define the project approach.

  b. Solution Outline phase

     i. Identify the requirements.

     ii. Define the high level architecture.

c. Macro Design phase

   i. Identify requirements.

   ii. Define the architecture.

d. Build the development environment

- Case Study 6: Degussa-Huls Divisions

  Team Assignment:

  From the results of your team's analysis of Degussa-Huls in Case studies 2 and 4, perform the first three phases of solution delivery for Degussa-Huls and then perform a project debriefing to gain insights and a better understanding of the phases and the application of the patterns approach.

  a. Solution Start-up phase

     i. Define the project approach.

  b. Solution Outline phase

     i. Identify the requirements.

     ii. Define the high level architecture.

  c. Macro Design phase

     i. Identify requirements.

     ii. Define the architecture.

     iii. Build the development environment.

3.6.3 Multiple choice

1. _____ is an activity in which the assets of the Patterns for e-business prove to be valuable.

   a. Identifying requirements.

   b. Documenting the Client Environment.

   c. Defining the solution.

   d. All of the above.

2. In the _____ activity during the proposal development stage, the navigation of the Patterns for e-business guides asset selection and offers insights into the overall business area and the main business drivers.

   a. Identifying requirements.

   b. Documenting the Client Environment.

   c. Defining the solution.

   d. All of the above.

3. In the _____ activity during the proposal development stage, the information gathered will help document the current IT environment of the client.

   a. Identifying requirements.

   b. Documenting the Client Environment.

   c. Defining the solution.

   d. All of the above.

4. In the _____ activity during the proposal development stage, the Patterns for e-business contribute significantly to the definition of the solution to be proposed.

   a. Identifying requirements.

   b. Documenting the Client Environment.

   c. Defining the solution.

   d. All of the above.

5. A benefit gained by usage of the Patterns for e-business during solution delivery is:

   a. Clarifying the applicable requirements that are used subsequently to drive the selection of the applicable patterns and assets.

   b. Delivering standard application and product configurations that can reduce the architecture-related development effort.

   c. Both A and B are correct.

   d. Neither A or B is correct.

6. A system context diagram illustrates interactions between the proposed system and:

   a. Users.

   b. Other systems.

   c. Both A and B are correct.

   d. Neither A or B is correct.

7. The focus of the _____ activity in the _____ phase is to familiarize the project team with the development method, assessing training and mentoring needs and tailoring the development method to the needs of the project.

   a. Defining the project approach, Solution Outline.

   b. Defining the project approach, Solution Startup.

   c. Defining the high level architecture, Solution Startup.

   d. None of the above.

8. The objective of the "Identifying Requirements" activity during the _____ Phase is to gather the overall business requirements and objectives, particularly concentrating on those that will have a significant impact on the solution.

   a. Solution Startup.

   b. Solution Outline.

   c. Solution Delivery.

   d. None of the above.

9. During the _____ phase, the overall shape and style of the system is decided based upon analysis of the key requirements.

   a. Solution Startup.

   b. Solution Outline.

   c. Solution Delivery.

   d. None of the above.

10. During the _____ phase, the team seeks to confirm the overall approach for the project as originally defined in the proposal.

   a. Solution Startup.

   b. Solution Outline.

   c. Solution Delivery.

   d. None of the above.

11. During the ____ phase of the project, the team may want to obtain more detailed requirements so that they can identify the actors and key business functions to be delivered by the solution.

   a. Solution Startup.

   b. Solution Outline.

   c. Solution Delivery.

   d. None of the above.

12. The main objective of the _____ activity in the ______ phase is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be paid to the more challenging areas of the proposed solution.

    a. Defining the project approach, Solution Outline.

    b. Defining the project approach, Solution Startup.

    c. Defining the high level architecture, Solution Startup.

    d. None of the above.

13. During Product mapping, the team will concentrate only on having an overview of the products that will support the overall solution. This is done to:

    a. Identify any gaps between the identified assets and what is needed to construct the solution.

    b. Identify integration points between the eliminated products to assess the effort involved in effectively achieving the solution.

    c. Both A and B are correct.

    d. Neither A or B is correct.

3.6.4 Fill in the blank

1. The term ____________ can be almost any person, or thing, provided the pattern fits the situation and the business process.
2. An important step in starting any project is to become familiar with any candidate __________ that may be useful during the development of the solution.
3. The purpose of the activity of _______________________________ is to put the hardware and software into place to support the development activities and any associated tools.
4. The purpose of the ____________ phase is to prepare for the build cycle(s) of a specific release of the system.
5. The main objective of the _____________________ activity during the Micro Design phase is to complete the design so that the build activities can be performed.

3.6.5 True or false

1. "Users" of a system can be almost any person, process, or thing, provided the pattern fits the situation and the user exercises the business process.
2. The objective of the "Identifying Requirements" activity during the Solution Startup Phase is to gather the overall business requirements and objectives, particularly concentrating on those that will have a significant impact on the solution.
3. The main objective of the defining the high level architecture activity in the macro design phase is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be paid to the more challenging areas of the proposed solution.
4. A salient feature of the Patterns is that there are a number of well understood and common approaches toward project management that function well across industries.
5. Once a Runtime pattern has been established, Product mapping can be defined, taking into account the actual IT environment of an organization.

3.6.6 Short answer

1. List the Solution Delivery Phases of the IBM Global Services Method.
2. What value is added by the IBM Patterns for e-business in the various Solution Delivery Phases of the IBM Global Services Method?
3. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Proposal Development activities, based on the case provided, please provide the following:
   a. What is the scope of the intended project based on the results of the Identifying Requirements activity?
   b. What are the Business Objectives relevant to the project? Of these objectives, which do you consider the most significant? Why?
   c. What are the key elements of the client environment relevant to the project? Of these elements, which do you consider the most significant? Why?

4. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Proposal Development activities, based on the case provided, please provide the following:

   a. Which Business Pattern was chosen? Why?

   b. Which Application Pattern was chosen? Why?

   c. Which Runtime Pattern was chosen? Why?

5. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Solution Delivery Phase: Solution Outline, based on the case provided, please provide the following:

   a. What is the overall objective of the "Identifying Requirements" activity?

   b. What additional non-functional requirements were identified during this activity?

6. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. Which application pattern was selected? Why?

7. In this chapter, the First Financial case was used to illustrate a solution based on a Single Business pattern solution. With the completion of the Macro Design Phase, based on the case provided, please provide the following:

   a. What is the overall objective of the "Identifying Requirements" activity?

   b. How does this differ from the same activity completed during the Solution Outline phase?

4

Building solutions on a multiple Business patterns basis

In this chapter, we explore how the Patterns for e-business can contribute to the development of a solution based on multiple Business and Integration patterns. It will be made clear that the Patterns for e-business are not a replacement for any development method in use during solution design or delivery, but are essentially an asset selection technique together with a rich set of associated assets that can be used during the phases/activities of a solution design/delivery project. In order to relate the contribution of the Patterns for e-business to the various phases and activities of solution design (project proposal) and solution delivery, the overall discussion has been framed within a high level description of the overall phase/activity structure of the e-business Custom engagement model of the IBM Global Services Method.

Outline

- Using the Patterns for e-business during proposal development
- Case study for proposal development
  - Identifying requirements
  - Documenting the client environment
  - Defining the solution
    - Business pattern(s)
    - Application pattern(s)
    - Runtime pattern(s)
    - Product mapping
  - Summary
- Using the Patterns for e-business during solution delivery
  - Solution Startup phase
  - Solution Outline phase
  - Macro Design phase
  - Micro Design phase
  - Build Cycle phase
  - Deploy phase
- Case study for solution delivery
  - Solution Startup phase
    - Defining the project approach
  - Solution Outline phase
    - Identifying requirements
    - ABC Insurance high-level business description
  - Defining the high level architecture
    - Identifying the actors and business functions
    - Connecting the actors and business functions
    - Evaluating business and IT drivers
    - Business patterns
    - Integration patterns
  - Identifying applicable application patterns
  - Self-Service application patterns

        - Business and IT Drivers for the Self-Service application patterns
    - Access Integration application patterns
        - Business and IT Drivers for the Access Integration application patterns
    - Runtime patterns
        - Runtime pattern - Self-Service::Router
        - Runtime pattern - Access Integration::Web Single Sign-on
        - Runtime pattern - Access integration::Personalized
- Macro Design phase
    - Identifying requirements
    - Defining the architecture
    - Building the development environment
- Micro Design phase
    - Refining the architecture
- Build Cycle phase
    - Building the production environment
- Deployment phase
    - Reviewing the solution plans
- Summary

Learning objectives

The objectives of this chapter help you to:

- Understand and explain a solution built on Multiple Patterns.
- Understand and explain the value added by IBM Patterns for e-business.
- Explain the reasoning behind the chosen solution.
- Explain the interactions between the chosen patterns.
- Explain the parts of the chosen solution and how those parts interact.

Real-world environment

In this section, we refer you to a real-world situation that uses the concepts in this chapter. The case study, called “NYPD changes the crime control equation by transforming the way it uses information” is available on the Web at:

http://www.redbooks.ibm.com/additional_materials/SG247466/NYPD_Changes_Crime_ControlEquation.pdf

4.1 Introduction

The process of developing e-business applications, based on a comprehensive set of requirements, requires a mature business solution delivery method. This chapter describes the iterative and incremental development method used to build an e-business application from requirements gathering down to deployment. The Patterns for e-business are used to provide reusable assets as we navigate from the problem domain to the solution domain within the development method.

We describe in detail how to build a solution on the basis of multiple Patterns for e-business, illustrating the implementation of a single coarse-grained application associated with one high-level use case.

The context for the discussion of how the Patterns for e-business influence the development of an e-business application is provided by the high level description of the phase/activity structure of the IBM Global Services Method. Throughout this chapter, particular attention is paid to those phases/activities that are most relevant for illustrating the value gained from incorporating the Patterns for e-business into our development approach. A comprehensive business case will be introduced and used as the basis for applying the Patterns for e-business.

As before, we will make a distinction between the use of the Patterns for e-business in a project definition/solution design context and the support it offers during the actual delivery of the project.

4.2 Using the Patterns for e-business during proposal development

Table 4-1 on page 87 gives an overview of some of the activities occurring during the development of a project proposal. These activities can take place in various contexts, such a during the preparation of a proposal linked to a particular Request for Proposal (RFP) or during the definition of a project proposal by an internal customer team. We will focus on the activities for which the assets of the Patterns for e-business are of most value, using ABC Insurance as a case study.

Table 4-1 Proposal development and the IBM Patterns for e-business

Proposal Development (Solution Design) activities	Added value of the IBM Patterns for e-business
Identifying requirements	The navigation of the Patterns for e-business asset catalog starts from a broad categorization of business requirements. Not only do these guide the asset selection process, but they also offer additional insights into the overall business area and the main business drivers.
Documenting the client environment	The information found in the request for proposal (RFP) or available to the customer team will help to document the current IT environment of the client. This information is key during the later stages of the asset selection process and at the same time creates an opportunity for a focused analysis of the request for proposal or the available documentation.
Defining the solution	Patterns for e-business contribute significantly to the definition of the solution to be proposed.

4.2.1 Case study for proposal development

We will now discuss each of the activities outlined in Table 4-1 in the context of the ABC Insurance case study.

Identifying requirements

In this section, we will introduce the business case for ABC Insurance.

Based on initial discussions with the CIO of ABC Insurance, the project team has developed a high level overview of the solution as follows.

ABC Insurance is an insurance provider, looking to provide electronic document submission capabilities to its independent agent community through the internet. While ABC Insurance expects the initial volume of electronic transactions to be small as compared to the volumes supported by its traditional business processes, they do believe that offering this infrastructure to their agent community ahead of the competition will provide growth opportunities in the future. ABC Insurance's customers will see this as a speed to market opportunity by reducing the time required to process insurance applications as well as a cost reduction strategy by electronically processing this information.

The solution will provide the following:

- A personalized user/agent experience
- Electronic insurance application submission
- Integration with existing systems for insurance application processing
- Access to insurance rating and underwriting information
- Access to insurance application status
- Electronic premium payment submission

ABC Insurance is also interested in providing these business functions through several channels, including Web browsers across the internet and personal digital assistants (PDAs).

Documenting the client environment

The key elements characterizing the customer's environment are as follows:

- The functionality will be delivered through a number of existing applications, each with its own authentication procedures. In order to simplify access for the insurance agents, an infrastructure enabling single sign-on will be implemented.
- A preference for Windows 2000/System x servers to host the Web infrastructure.
- An installed base of either Windows 2000/System x or System z.
- Existing back-end applications deployed on System z.

Defining the solution

Based on the information gathered from ABC Insurance (and summarized in the previous sections), we decide to use the Patterns for e-business for a fast path definition of a suitable and proven solution meeting the business requirements of ABC Insurance. This fast path approach is only expected to provide an 80% fit for the full set of ABC Insurance's requirements so a comprehensive development method will still be needed to customize the selected assets to result in a 100% fit.

These patterns should be considered candidate assets that leverage the experience of architects to create solutions quickly, whether for small local businesses or large multinational enterprises. Specifically in the context of the development of a project proposal, the ability to quickly translate requirements into a coherent and viable solution is extremely valuable. Also, the proven product configurations documented in the Patterns for e-business asset catalog contribute significantly to producing a high quality project proposal on time.

The key elements contributed by the Patterns for e-business are:

- Identification of required functionalities
- Product selection guidance
- Information regarding the interactions between products

For the purposes of the proposal, the following four-step procedure is used to arrive at the recommended solution:

1. Selection of the Business pattern(s):

   An initial analysis of the high-level business requirements described above indicates that the following Business patterns may be applicable to this solution:

   – Self-Service business pattern

     The Self-Service business pattern is applicable if the requirements suggest that users need to interact directly with existing data. In this particular case, ABC Insurance is interested in providing access to its existing systems across the Web to its insurance agents.

   – Access Integration pattern

     The Access Integration pattern describes recurring designs that enable access to one or more Business patterns. In particular, this pattern enables access from multiple channels (devices) and integrates the common services required to support a consistent user interface. In ABC Insurance's case, the pattern is relevant due to the requirement for supporting the agents efficiently in their work by supplying them with a single, consistent, and seamless access mechanism to the various applications.

2. Selection of the Application pattern(s):

   The Application pattern focuses on the application level functionality required to support the identified Business pattern(s).

   Based on the available information, the following Application patterns are selected.

   – Self-Service: Router application pattern

     ABC Insurance needs to make the functionality of a number of its existing back-end systems available across a number of different channels (initially only across a Web channel, but a PDA has been identified as a future extension). The Router application pattern provides a structure for applications that require the intelligent routing of requests from multiple delivery channels to one of multiple back-end applications.

- Access Integration: Web Single Sign-on application pattern

  Agents will need to access multiple back-end systems and a consistent security policy needs to be enforced across all of these applications. Putting in place a single sign-on facility will simplify the user's interactions with the system and ease the administrative overhead.

3. Selection of the Runtime pattern(s):

   The Runtime pattern focuses on the technical functionality needed to support an Application pattern.

   - Self-Service: Router Runtime pattern

     In this Runtime pattern, the functions of the router tier (the intelligent routing of requests from multiple delivery channels to one of multiple back-end applications) are delivered by an integration server.

   - Access Integration: Web Single Sign-on Runtime pattern

     A number of variants exists, each with their own assumptions. Given ABC Insurance's requirements, the applicable variant of this Runtime pattern is the one in which applications rely on the security services of the underlying application server.

4. Obtain the Product mapping:

   Given the existing base of Windows 2000/System x servers, the corresponding Product mapping is selected.

   This Product mapping immediately gives us an overview of the required products and their relationships.

   Based on the information obtained from the Patterns for e-business, the proposal can then be further developed by:

   - Estimating the man days needed to build new functionality or to integrate existing functionality
   - Estimating the software budget, taking into account among others the suggested software products

   Note: Other factors are crucial in being able to realistically budget for the proposed solution, like sizing estimates, available skills, and other relevant non-functional requirements (availability, performance, and so on), but these are outside the scope of this discussion.

   - Specifying the resource profiles required to deliver the solution (among others derived from selected products)
   - Specifying the tooling required for development (among others derived from selected products)

4.2.2 Summary

This section showed how multiple patterns from the Patterns for e-business catalog can be used very effectively in defining (a part of) the solution in response to a request for proposal or in a project definition context. The Patterns for e-business allow us to arrive very quickly at such a solution through an asset selection process starting from an analysis of the main business and IT drivers present in the request for proposal. The obtained results, expressed in terms of required software products, can be used immediately to drive the budgeting aspect of the solution design and will increase the overall technical quality of the proposal given its reliance on the proven software and hardware product combinations found in the Patterns for e-business asset catalog.

4.3 Using the Patterns for e-business during solution delivery

In Table 4-2, we take a closer look at the phases and activities of the e-business Custom engagement model. For each of the phases, we identified those activities where the Patterns for e-business can be leveraged. In this context, the Patterns for e-business should not be viewed as a replacement for a method, but as a complementary technique to be used within the context of a method.

Table 4-2 provides references to additional IBM Redbooks publications and the Patterns for e-business Web site to assist in a given activity.

Table 4-2 IBM Global Services Method and IBM Patterns for e-business

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Phase: Solution Startup	
Defining the project approach	Starting from the approach outlined in the project proposal, confirm and refine the proposed approach based on any additional information. This is usually done both with the project team and client personnel. The use of the IBM Patterns for e-business will be taken into account during this stage since this will help to shape the overall approach.

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Phase: Solution Outline	
Identifying Requirements	The focus of this activity is to gather the overall business requirements and objectives while concentrating on those that will have a significant impact on the solution. One technique that supports getting an overall view of the solution's requirements involves the development of a system context diagram. This diagram represents the proposed solution emphasizing the interactions to be supported with its environment, both in terms of people and systems. The Business patterns can provide richness to this diagram by relating it to similar trends in other industries while at the same time allowing for the validation of the selected pattern(s).
Defining the high level architecture	The main objective of this activity is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be given to the more challenging areas of the proposed solution. Based on the architecturally significant requirements, the architect will validate the business and IT drivers from the client and match these against the ones associated with the patterns. This process will then lead to the identification of the involved Application patterns. The applicable Application patterns can be documented as part of an architecture overview diagram that would feature, for example, the Router Application pattern. An adopted Application pattern will lead the architect to a specific set of Runtime patterns and Product mappings taking into account the actual IT environment of the customer. It is very important that at this stage, an accurate assessment be made of any gaps between the identified assets and the actually required solution so that adequate attention can be given to those areas through either the selection of additional products or through custom development.

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Phase: Macro Design	
Identifying requirements	The focus of this activity is to gather and refine all applicable requirements (instead of focusing only on the architecturally significant ones) so that the results from the “Identifying requirements” activity performed during the Solution Outline phase can be confirmed and further refined.
Defining the architecture	This activity will focus on specifying the overall architecture, both in terms of how it addresses the functional and non-functional requirements so that in subsequent phases, a start can be made on its implementation. During this activity, the architecture defined during the Solution Outline phase will be confirmed and further refined. This time, we will review the patterns selected during the previous phase based on all known requirements. Once the Business patterns have been confirmed, the following steps can be taken: 1. Finalize the Application patterns to be used. 2. Select the Runtime pattern for each of the identified Application patterns. 3. After selecting the Runtime patterns, an experienced architect should integrate these patterns into the overall architecture. 4. Perform the Product mapping on the basis of the current IT environment of the customer. For the selected Runtime patterns, review the Product mappings on the patterns Web site at: http://www-106.ibm.com/developerworks/patterns/u2b/at5-product-map.html This information can be used as a guideline for selecting the products in case these have not yet been defined. In situations where the client has already predefined an infrastructure, the information on this Web site can be used to validate the decisions that have been made, again based on the previously identified Business, Integration, Composite, Application, and Runtime patterns.

Solution Delivery Phases	Added value of the IBM Patterns for e-business
Building the development environment	Review the application development chapters of the applicable pattern IBM Redbooks publications to support the definition of the development environment. Additionally, review product installation instructions in the applicable pattern IBM Redbooks publications to assist with the product installs for the development environment.
Phase: Micro Design	
Refining the architecture	The main objective of the micro design is to complete the design so that the build activities can be performed. Guidance on recommended design practices can be found in the application design and application development chapters of the applicable pattern IBM Redbooks publications.
Phase: Build Cycle	
Building the production environment	The focus of this activity is to build the production environment based on the architecture defined during the previous phases. Again, refer to the product installation instructions in the applicable pattern IBM Redbooks publications to assist in building the production environment. This may be especially helpful if there are differences between the development and test and production environments.
Phase: Deploy	
Reviewing the solution plans	Review the system management guidelines chapters in the applicable pattern IBM Redbooks publications.

4.4 Case study for solution delivery

Now we begin the solution delivery for ABC Insurance.

4.4.1 Solution Startup phase

During this phase, we will seek to confirm the overall approach for the project as originally defined in the proposal. An important input to this activity is the information previously made available by ABC Insurance and the actual solution defined during the Project Proposal stage. Particularly relevant for this phase is the information related to the possible use of assets during the engagement.

For the purposes of this section, we will assume that the problem setting introduced in the proposal development section is still valid.

Defining the project approach

The focus of this activity lies in familiarizing the project team with the development method, assessing training and mentoring needs, and tailoring the development method to the needs of the project taking the use of assets into account.

An important step in starting any project is to become familiar with any candidate assets that may be of assistance during the development of the solution. Using candidate assets can also improve the profitability of a project as well as lower its risk given that the solution will be based on assets used in previous successful implementations. The candidate assets we are particularly interested in are the Patterns for e-business. Based on the information gathered from the CIO at ABC Insurance, we should be able to determine whether the Patterns for e-business are applicable to this solution.

4.4.2 Solution Outline phase

During the Solution Outline phase, the overall shape and style of the system is decided based upon an analysis of the key requirements. The emphasis during this phase lies much more in identifying all the required functionalities than on exploring each one in detail. The overarching objective is to get an accurate view of the overall effort required to deliver the system. Consequently, particular attention will be paid to those areas deemed to be particularly challenging.

Identifying requirements

Starting from the requirements expressed by ABC Insurance and the way these were interpreted in the project proposal, we will reexamine the customer's requirements in more detail.

If this initial information is not available, then some of the steps normally taken during the project proposal development will be performed at this time, for example, identifying the high level requirements.

During this phase of the project, we want to obtain more detailed requirements so that we can identify the actors and key business functions to be delivered by the proposed solution. This information is usually obtained from the high-level use cases; however, for the purposes of this discussion, we will use the high-level business description noted below.

We first define some frequently used terms:

- An actor represents a coherent set of roles. Several users can play the same role and one user can perform several roles. Examples of actors could include a Customer Service Representative (human), and the Credit Authorization Service (external system).
- A business function represents a high-level specification extracted from the high-level business requirements. For example, a portal with a personalized UI would be considered a business requirement, and a secured logon would be considered a business function supporting this requirement.

The actors and business functions obtained from the high-level business description will serve as the basis for developing several deliverable during the Solution Outline, which will be further refined during the Macro Design phase. The high-level business description for ABC Insurance was developed after a number of work shops with the client IT personnel and the CIO.

To clearly identify the requirements embedded in the following business description, we will underline the actors and the items in **bold** will represent the business functions.

ABC Insurance high-level business description

The ABC Insurance solution will provide a self-service solution for our agents. The solution must support a browser-based user interface (UI) across the internet. Our future plans are to expand this interface to wireless pervasive devices, such as PDAs. The solution should provide a personalized experience to the agent, at a minimum covering requirements such as a **secured logon** and a **personal home page**. For example, this home page will be customized to a specific agent's requirements, such as pending applications and upcoming renewals.

The ABC Insurance solution will allow agents to **submit applications** for insurance coverage in an electronic document format. To accomplish this, the solution must interface with the underwriting and rating existing applications. The solution should provide **work in progress** capabilities, indicating the status of a submitted application. An **alert feature** should be included as part of the work in progress feature, prompting the agent for updates or missing information in the application.

The solution should accommodate **electronic premium payments** for approved applications. Links to the ABC Insurance financial applications will be required to accommodate this feature. ABC Insurance uses First Financial Bank for processing the electronic payments.

> **Note:** For more information about the high-level business description, refer to Chapter 2 of the book *Patterns for e-business: A Strategy for Reuse*, by Adams, et al.

Using the high-level business description as a guide, we want to develop a system context diagram. By describing the actors, the system to be delivered and their interactions, a system context diagram helps drive out a precise definition of the scope of the system. Examples of such actors can be service providers, regulatory agencies, financial institutions, and so on.

The system context diagram varies from client to client depending on the requirements, the current IT environment, and the type of solution to be delivered. For example, the system context diagram for ABC Insurance may look like the one shown in Figure 4-1, where the Web-Enabled Electronic Document Processing Gateway represents the system to be delivered.

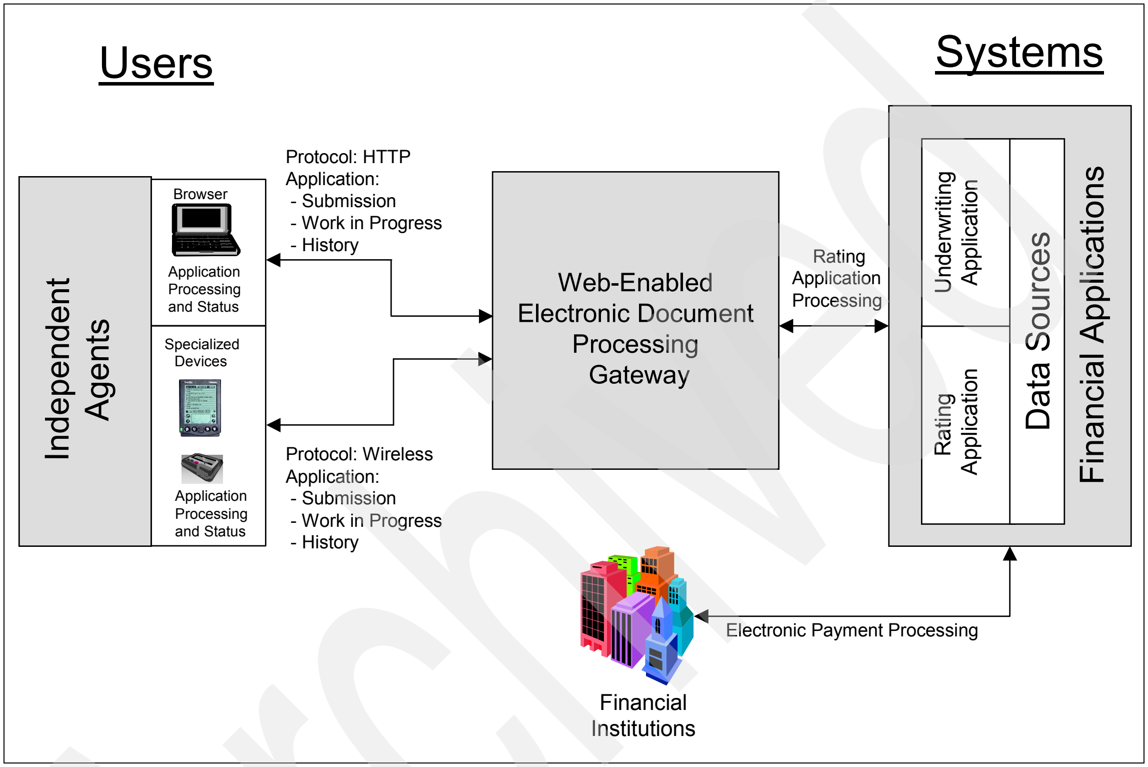


Figure 4-1 ABC Insurance System Context diagram

Defining the high level architecture

The main objective of this activity is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be paid to the more challenging areas of the proposed solution.

Using the high-level business description as a guide, we develop an architecture overview diagram. Whereas the system context diagram emphasizes how the system to be delivered fits into its environment (in terms of the interactions, it needs to support various actors), the architecture overview diagram (AOD, also known as a Solution Overview Diagram, as described in the *Patterns for e-business: A Strategy for Reuse* book) gives an overview of the system that will be delivered, that is, it shows the major building blocks of the solution itself. Such a representation is ideally suited to overlay the Patterns identified thus far, in this case the Business and Integration patterns. In fact, the architecture overview diagram will serve as the basis for depicting all the Patterns that we leverage for developing the solution for ABC Insurance.

The structure of the remainder of this section will reflect the approach followed in determining the patterns applicable to our case study:

- Identifying the actors and business functions
- Connecting the actors and business functions
- Evaluating business and IT drivers
- Identifying applicable Application patterns
- Identifying applicable Runtime patterns
- Integrating the applicable Runtime patterns

Identifying the actors and business functions

Using the high-level business description, we develop a visual representation of the actors and business functions shown in Figure 4-2. This figure shows both actors (the underlined items taken from the high-level business description above) as well as business functions (the bold items taken from that same description). Existing applications have been represented with double lines on the box.

.

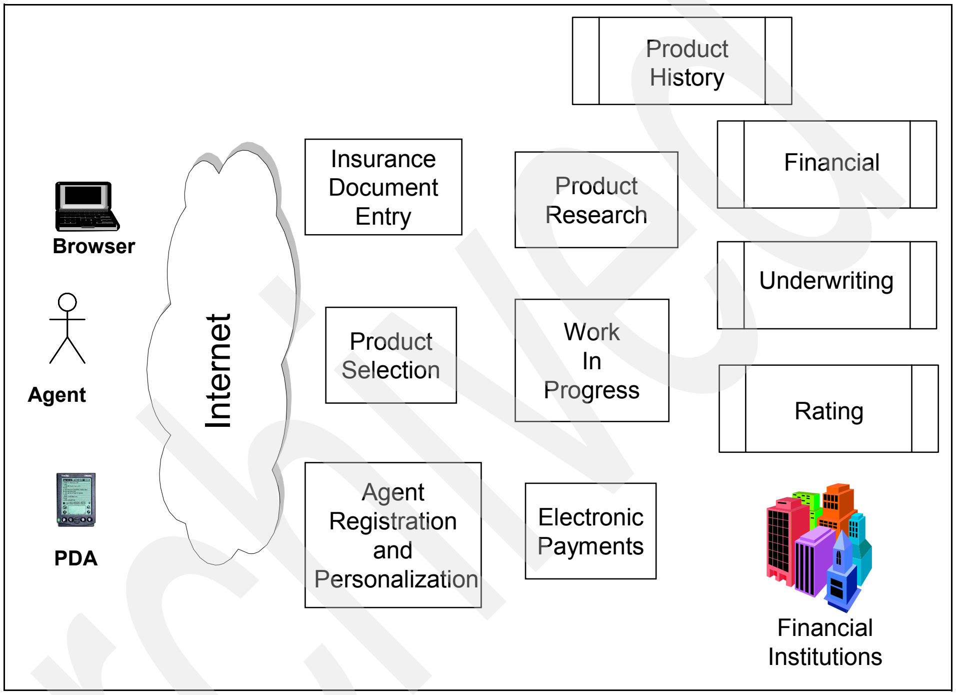


Figure 4-2 ABC Insurance actors and business functions

Connecting the actors and business functions

Using the high-level business description, we want to link up each business function with either:

- Another business function
- An actor (either an existing system or a human)

The resulting architecture overview diagram is shown in Figure 4-3 on page 101.

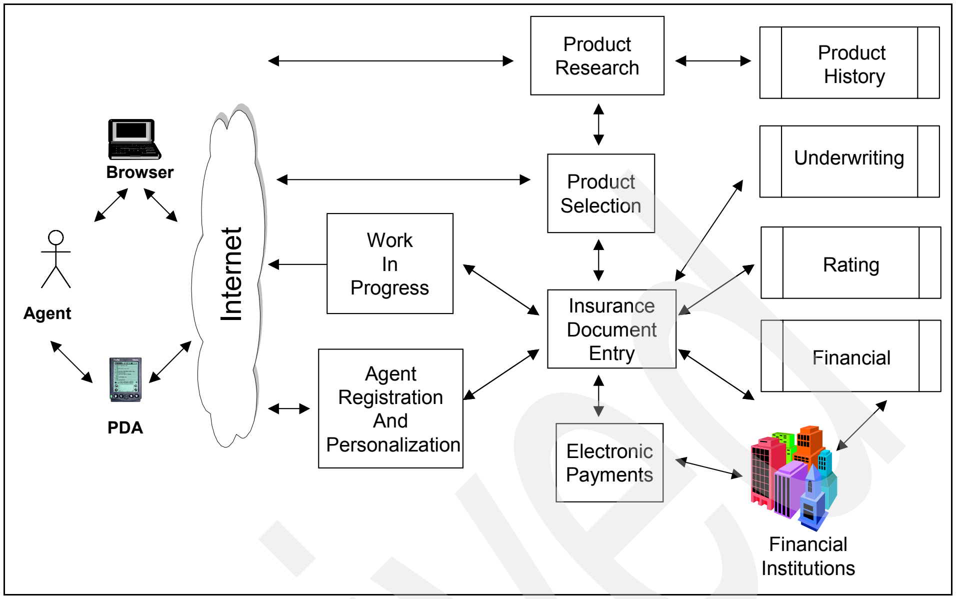
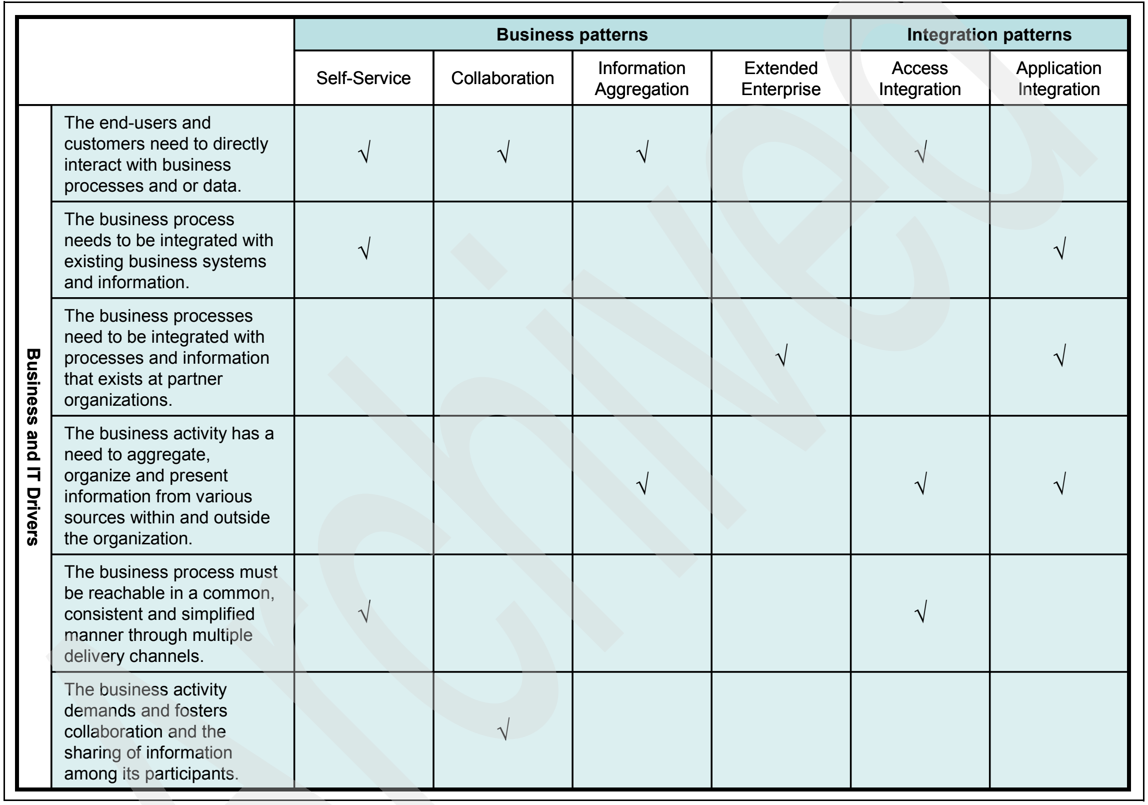


Figure 4-3 Architecture overview diagram with business processes

Note: Typically, different architecture overview diagrams are produced for different target audiences. For example, in developing a solution where Patterns are not used, the architect may choose a more technical representation of the system to be delivered. In projects where Patterns are leveraged, the architecture overview diagram will take on a more "business view" as is the case within this chapter and demonstrated in the diagram shown above.

Evaluate business and IT drivers

The diagrams in Figure 4-2 on page 100 and Figure 4-3 on page 101 provide the initial structure of the system on which we can begin to overlay the identified Business and Integration patterns. To accomplish this, we must analyze their associated business and IT drivers. The business and IT drivers are a consistent set of objectives that assist the architect in selecting the appropriate Business and Integration patterns. In Figure 4-4, we will summarize the Business and IT drivers for the Business and Integration patterns.



		Business patterns				Integration patterns	
		Self-Service	Collaboration	Information Aggregation	Extended Enterprise	Access Integration	Application Integration
Business and IT Drivers	The end-users and customers need to directly interact with business processes and or data.	√	√	√		√	
	The business process needs to be integrated with existing business systems and information.	√					√
	The business processes need to be integrated with processes and information that exists at partner organizations.				√		√
	The business activity has a need to aggregate, organize and present information from various sources within and outside the organization.			√		√	√
	The business process must be reachable in a common, consistent and simplified manner through multiple delivery channels.	√				√	
	The business activity demands and fosters collaboration and the sharing of information among its participants.		√				

Figure 4-4 Business and IT drivers for the Business and Integration patterns

The adopted approach that is based on an analysis of the Business and IT drivers is applicable both to Business and Integration patterns. In both cases, it is based on correctly identifying what the client is trying to achieve. In this case, the high level business description may be used as a starting point and the embedded requirements may then be matched against the business and IT drivers in Figure 4-4 to determine the applicable Patterns for this solution. In other situations where the solution may be more complex or more information is available, it may be necessary to perform more analysis to determine the applicable Patterns.

Based on the information provided in the business scenario, we can quickly determine if any Business or Integration patterns will be applicable to this solution. In a typical project, Composite patterns and Custom designs would also be reviewed as candidate assets.

We will first revisit the Business patterns and perform a more refined analysis than the one carried out initially at the project proposal stage. Our assessment of relevant patterns runs as follows.

Business patterns

Our analysis of the business scenario indicates that the following Business patterns are applicable:

- Self-Service

  In this case, ABC Insurance is interested in creating a personalized experience for their customers or agents. The Self-Service Business pattern (also known as user-to-business) fits here given that the agent is a user of ABC Insurance's business systems.

- Collaboration

  The Collaboration pattern fits with this business scenario given the requirement for being able to access the insurance application status. This implies an interactive (collaborative) environment between the agent and ABC Insurance personnel.

- Information Aggregation

  ABC Insurance is interested in providing rating and underwriting information back to the agent community. This will require product history data to be aggregated and distilled into product research data for each agent.

- Extended Enterprise

  ABC Insurance is interested in electronic payment processing. In most cases, this will involve an external payment processing center like a financial institution.

A visual depiction of the applicable Patterns for the ABC Insurance business case is shown in Figure 4-5.

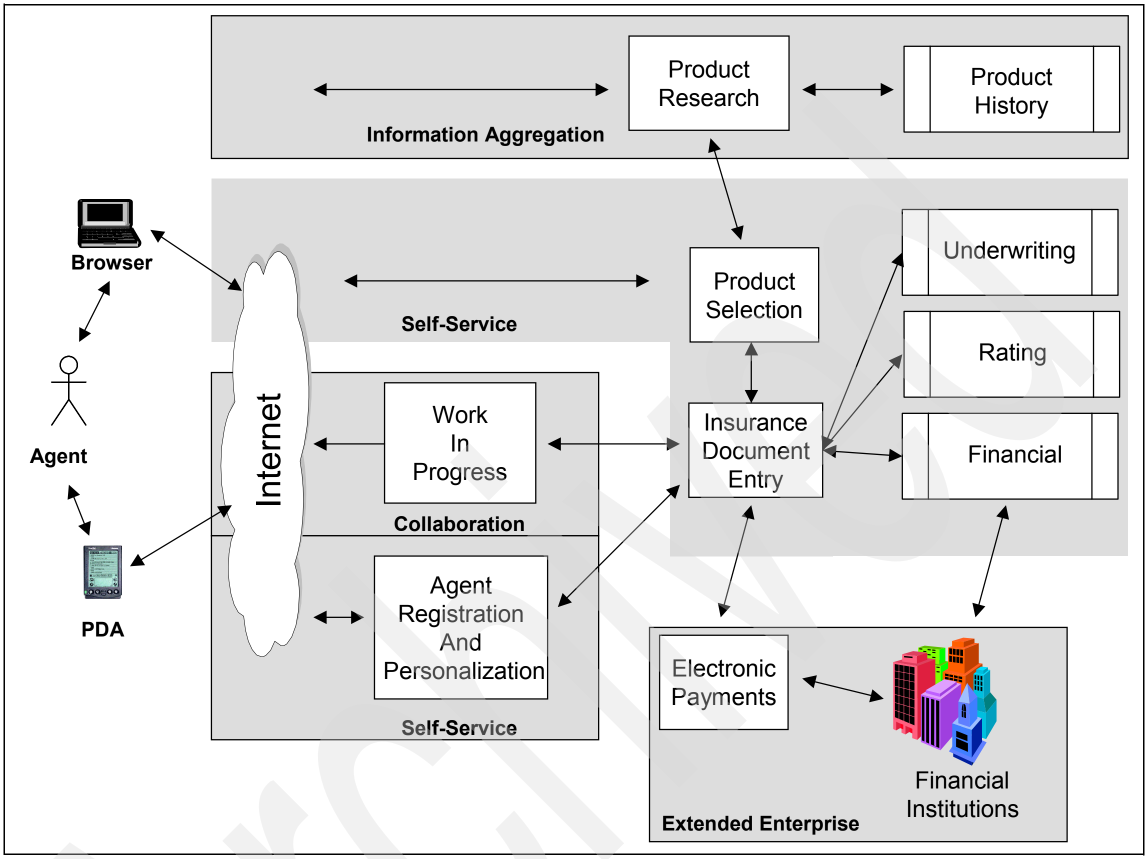


Figure 4-5 Applying the Business patterns to the architecture overview diagram

Integration patterns

After reviewing the high-level business scenario, it appears that the following Integration patterns are applicable for this solution:

- Access Integration

  The Access Integration pattern provides for front-end integration of multiple services and information through a common portal. The pattern is responsible for handling multiple client device types, single sign-on, personalization, and providing a common look and feel for the application interfaces. As described in the business scenario, many of these requirements are key to developing the proper solution for ABC Insurance.

- Application Integration

  The Application Integration pattern provides for the seamless back-end integration of multiple applications and data without the user accessing them directly. The new Web-based tier that will be developed to support this solution will need to access ABC Insurance's existing systems and data. As such, this pattern will be important during solution development.

In Figure 4-6, we show the Integration patterns we identified based on our analysis. In this figure, each individual Access Integration and Application Integration pattern is not illustrated. We realize that we need a number of Application Integration patterns, but to keep the diagram simple we do not show these. For readers interested in a more thorough treatment, refer to *Patterns for e-business: A Strategy for Reus*e by Adams, et al. We will therefore implement this simplication by using one Business pattern (Product Selection, Self-Service Business pattern) and one Access Integration pattern, and combining them.

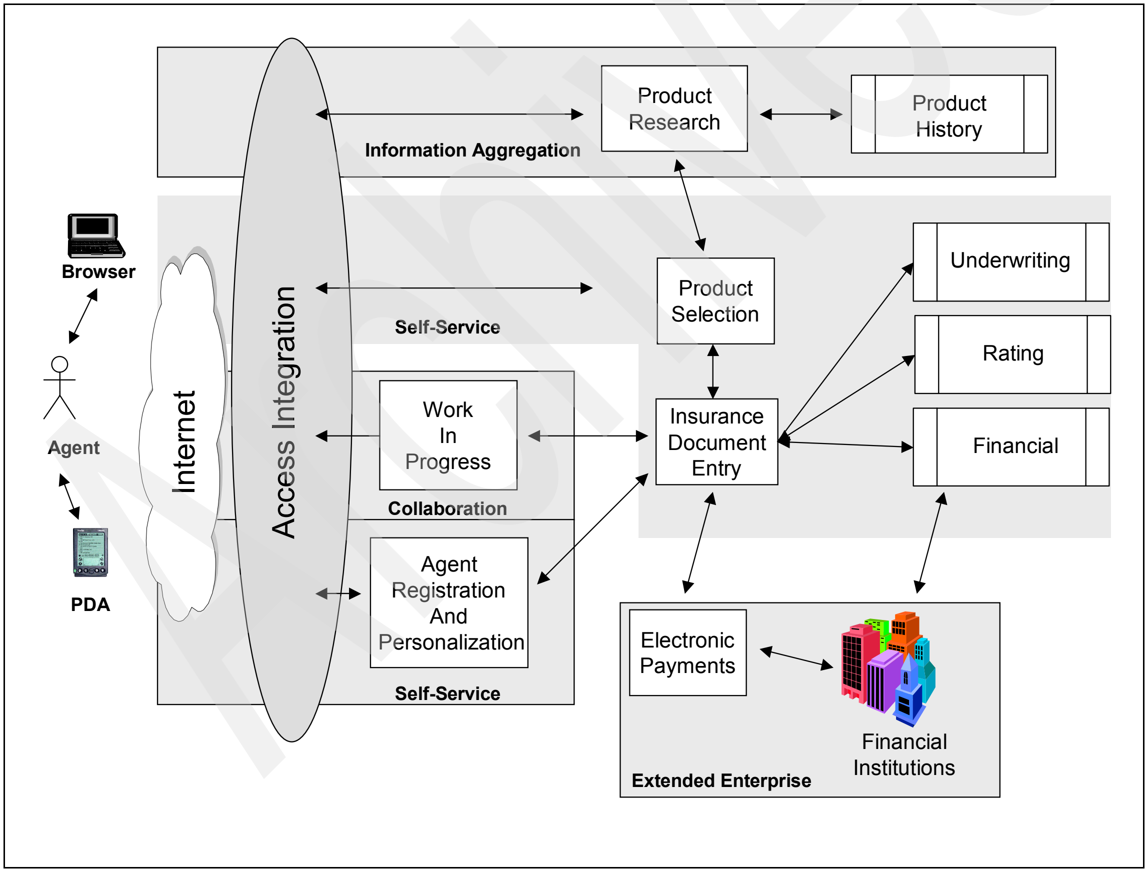


Figure 4-6 Applying the Integration patterns to the architecture overview diagram

Identifying applicable Application patterns

As noted in Figure 4-5 on page 104 and Figure 4-6 on page 105, we have identified two Self-Service business patterns (focusing only on Product Selection) and one Access Integration pattern. We must now identify the Application patterns for each.

A preparatory approach that can be used is to take a look at the Enterprise Application Integration (EAI) architecture and determine the types of application integration topologies that apply. For example, one topology may suggest a point-to-point application integration architecture while another a hub-and-spoke architecture.

Analyzing the different topologies can be helpful in selecting the correct Application patterns for a given solution. While this chapter does not discuss an EAI-based architecture for ABC Insurance, we can suggest that the solution we are developing is a component of their overall EAI architecture. Given the information we have thus far and after analyzing the architecture overview diagrams in Figure 4-5 on page 104 and Figure 4-6 on page 105, we can see that a hub-and-spoke topology may be more appropriate for this solution.

Point-to-Point suggests either single or multiple access channels accessing a single or common existing applications. This solution, however, has multiple access channels accessing multiple existing systems, such as underwriting, rating, and financial, thus suggesting that a hub-and-spoke topology may be more appropriate for this solution. Keeping this in mind, we will now take a look at the Business and IT drivers for the Application patterns.

The standard approach to identifying the appropriate Application patterns is to compare the key business requirements gathered so far to the Business and IT drivers for the Self-Service and Access Integration Application patterns.

Self-Service Application patterns

Business patterns allow us to categorize the interactions between external parties and internal business processes supported by applications and data. Further development now requires that we identify the appropriate Application patterns to support these Business patterns, in this particular case, the Self-Service Business pattern.

Business and IT Drivers for the Self-Service application patterns

Figure 4-7 on page 107 and Figure 4-8 on page 108 show the existing Self-Service application patterns and the associated Business and IT drivers. For ABC Insurance, we need to determine which Application pattern would fit best based on their requirements. It is important to keep in mind that a hub-and-spoke topology seems to be the best fit for this particular solution. With this in mind, we will now take a closer look at the business and IT Drivers for these Application patterns to further narrow down the selection.

		Self-Service Application patterns						
		Stand-Alone Single Channel	Directly Integrated Single Channel	As-Is Host	Customized Presentation to Host	Router	Decomposition	Agent
Business Drivers	Decrease the time to market.	√		√	√			
	Improve the organizational efficiency.		√	√	√			
	Reduce the latency of business events.		√			√	√	√
	Enable easy adaptation during mergers and acquisitions.					√	√	√
	Integrate across multiple delivery channels.					√	√	√
	Provide a unified customer view across lines of business.						√	√
	Support effective cross-selling.							√
	Support mass customization.							√

Figure 4-7 Business drivers for the Self-Service Application patterns

		Self-Service Application patterns						
		Stand-Alone Single Channel	Directly Integrated Single Channel	As-Is Host	Customized Presentation to Host	Router	Decomposition	Agent
IT Drivers	Decrease the time to market.	√			√			
	Improve the organizational efficiency.			√	√	√	√	√
	Reduce the latency of business events.		√	√	√	√	√	√
	Enable easy adaptation during mergers and acquisitions.		√	√	√	√	√	√
	Integrate across multiple delivery channels.		√	√	√	√	√	√
	Provide a unified customer view across lines of business.					√	√	√
	Support effective cross-selling.					√	√	√
	Support mass customization.					√	√	√

Figure 4-8 IT Drivers for the Self-Service Application patterns

One of the primary Business and IT Drivers for ABC Insurance is the ability to integrate across multiple delivery channels, in this case, a browser and pervasive devices. They also are interested in minimizing the complexity of the enterprise and in future growth aspects of this application. ABC Insurance, as part of its EAI strategy, is interested in expanding their business and possibly taking on additional insurance carriers in the future in an attempt to offer a broader range of products to its agent base. Based on these additional requirements, it would seem that the Router, Decomposition, and Agent application patterns would be the best fit for ABC Insurance. In addition, these three Application patterns support the hub-and-spoke topology as discussed earlier; each, however, extends this topology by providing additional capabilities. The following is a more in-depth discussion of these Application patterns.

- Router

  The Router Application pattern supports multiple delivery channels connecting to one or more back-end existing applications through a router tier. This router tier can route a single message to one or more applications within the same business process. For example, ABC Insurance is interested in processing insurance applications across a single line of business, in this case let us say Property and Casualty. The router tier can process a single request within this business line and access the required existing systems to produce a response, in this case the rating and underwriting systems.

- Decomposition

  The Decomposition Application pattern extends the hub-and-spoke architecture provided by the Router Application pattern. It decomposes a single, compound request from a client into several, simpler requests and intelligently routes them to multiple back-end applications. Typically, the responses from these multiple back-end applications are recomposed into a single response and sent back to the client. This Application pattern is especially useful if the solution is supporting multiple business lines. For example, as noted before, the solution for ABC Insurance addresses the requirements of their Property and Casualty line of business. If they were to expand their business to underwrite Auto and Health insurance policies and they wanted to incorporate theses new business lines into this solution, the Decomposition Application pattern would be a good fit given that a single request may have to access multiple existing systems for multiple lines of business. The request would then need to be reconstructed into a single response. As noted in the requirements from the high-level business description, this is currently not the case at ABC Insurance although they do expect to expand to additional lines of business in the near future.

- Agent

  The Agent Application pattern structures an application design that provides a unified customer-centric view that can be exploited for mass customization of services and for cross-selling purposes. While ABC Insurance does intend to provide a customized view to its agents, they do not yet offer multiple lines of business; therefore, the need to leverage these in a cross-selling capacity is not yet established. The Router Application pattern will support the personalization requirement.

In reviewing the additional requirements and the functionality provided by each of the Application patterns, it would appear that the Router Application pattern would be the best fit, as well as offer the least complexity for this particular solution. As such, we will use the Router Application pattern as the basis for identifying and applying the Runtime patterns and Product mappings.

Access Integration Application patterns

The Access Integration pattern gives users a single, consistent, and seamless access mechanism to various applications that would otherwise require the use of several different access mechanisms. This Integration pattern is useful when:

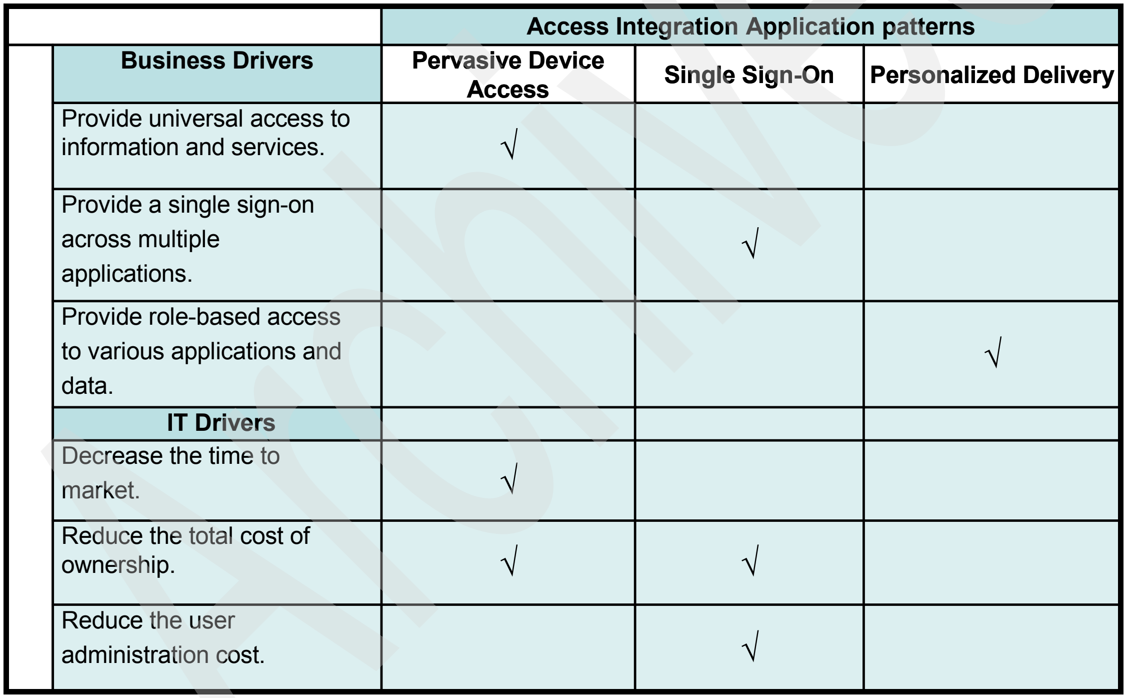
- Users need access to multiple applications and information sources without every application requiring its own sign-on to establish a separate security context.

- These applications need to be accessed using multiple devices such as fat clients, browsers, voice response units, mobile devices, and PDAs.
- A common look and feel is required for all applications.
- The user wishes to customize the choice of applications and how they are presented.

We now need to identify the appropriate Application patterns to support the Access Integration pattern.

Business and IT drivers for the Access Integration application patterns

The Business and IT Drivers for the Access Integration application patterns are shown in Figure 4-9.



	Access Integration Application patterns		
Business Drivers	**Pervasive Device Access**	**Single Sign-On**	**Personalized Delivery**
Provide universal access to information and services.	√		
Provide a single sign-on across multiple applications.		√	
Provide role-based access to various applications and data.			√
IT Drivers			
Decrease the time to market.	√		
Reduce the total cost of ownership.	√	√	
Reduce the user administration cost.		√	

Figure 4-9 Business and IT drivers for the Access Integration Application patterns

Note: The primary business driver for choosing the Personalization Delivery Application pattern is to increase usability and improve the efficiency of Web applications by tailoring their presentation to the user's role, interests, habits or preferences.

ABC Insurance is interested in providing multiple channel access to its agent community through a single secure sign-on, as noted in the high level business description. This will allow the agent to access multiple existing systems for insurance rating and underwriting purposes. ABC Insurance is also interested in providing a personalized view to its agents.

We examine each of the Access Integration Application patterns in more detail to determine the best fit for ABC Insurance. They are as follows:

- Pervasive Device Access Application pattern

  The Pervasive Device Access Application pattern brings a new tier into the architecture. This tier is responsible for the pervasive extensions to the original application. The function of this tier is to convert the HTML issued by the application presentation logic into a format appropriate for the pervasive device. In this way, the Pervasive Device Access Application pattern provides a structure for extending the reach of individual applications from browsers and fat clients to pervasive devices such as PDAs and mobile phones.

  This pattern can be leveraged in developing the solution for ABC Insurance. However, ABC Insurance has indicated that using PDAs as a delivery channel is a future requirement and not within scope of the immediate solution. Therefore, we will not include this Access Integration Application pattern at this time.

- Web Single Sign-On Application pattern

  The primary business driver for choosing this Application pattern is to provide seamless access to multiple applications with a single sign-on while continuing to protect the security of enterprise information and applications. This Application pattern satisfies one of ABC Insurance's requirements and will be used in our analysis.

- Extended Single Sign-On Application pattern

  This Application pattern takes the Single Sign-On Application pattern and extends it to include security for the back-end systems. For solutions with strong privacy or audit requirements, this approach is needed. The requirements for ABC Insurance do not really dictate such strong security requirements for their back-end systems. As such, this Application pattern will not be used.

- Personalized Delivery Application pattern

  The Personalized Delivery Application pattern provides a framework for giving access to applications and information tailored to the interests and roles of a specific user or group. This pattern extends basic user management by collecting rich profile data that can be kept current up to the user's current session. Data collected can be related to application, business, personal, interaction, or access device-specific preferences. ABC Insurance is interested in creating a personalized experience for their agents. These requirements will mainly support the business functions, such as work in progress activities. This Application pattern will be useful given these requirements and therefore will be used in developing the solution.

Our analysis of Access Integration Application patterns and ABC Insurance's requirements support using the Single Sign-On and Personalized Delivery Application patterns. Again, the Pervasive Device Application pattern is applicable; however, given that it is a future requirement, it will not be discussed further in this IBM Redbooks publication.

Runtime pattern - Self-Service: Router

In the Router Application pattern, the router tier serves as an integration point for delivery channels in the presentation tier, allowing access to multiple back-end applications. In this Runtime pattern, the functions of the router tier are performed by an integration server. Figure 4-10 on page 113 shows a Runtime pattern for Self-Service: Router.

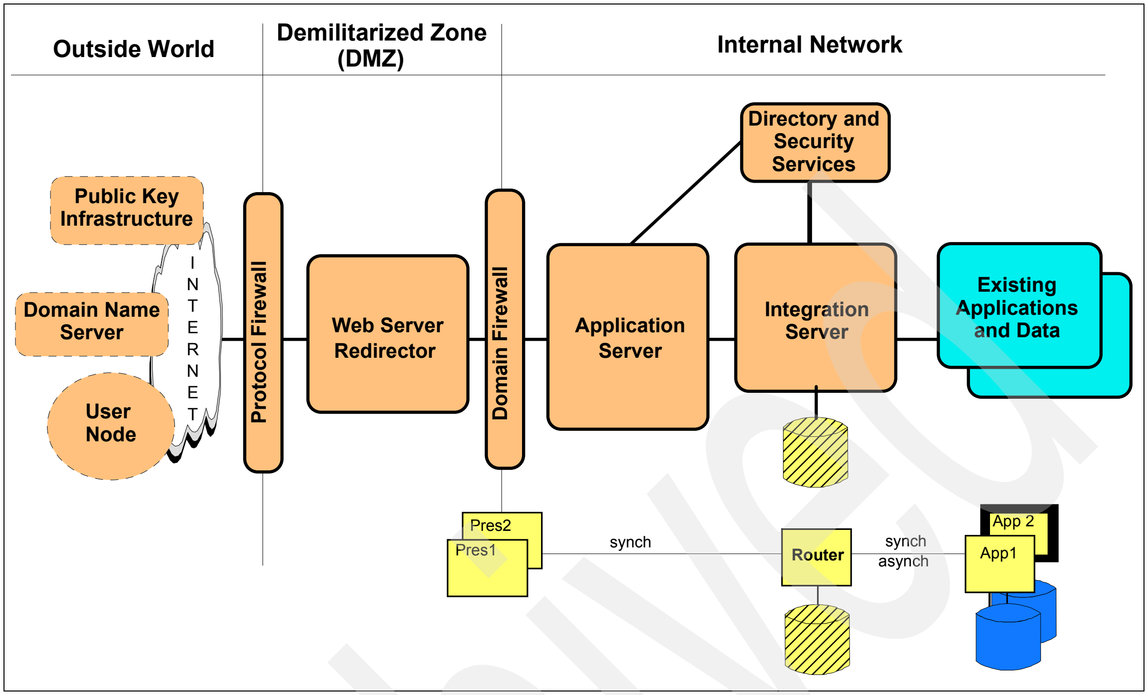


Figure 4-10 Runtime pattern - Self-Service: Router

Each of the nodes are described in detail on the Patterns for e-business Web site. Refer to the following URL for more information:

http://www-106.ibm.com/developerworks/patterns/u2b/at5-runtime.html

Runtime pattern - Access Integration: Web Single Sign-on

There are two variations of the Runtime pattern for Web Single Sign-On: one where a homogeneous application server is used and the other where a heterogeneous application server is used. The basic difference between these two approaches comes down to which server is performing the user authentication and security checks. In a homogeneous environment, all applications can exploit the application's security server for single sign-on functionality. In the heterogeneous environment, a separate security server is used. This is also used in environments where you want to support single sign-on to multiple back-end platforms.

The mid-tier environment for ABC Insurance will be on a single platform connecting to existing systems that exist on one platform as well, resulting in a low level of complexity. Therefore, the homogeneous application server approach is appropriate for this solution.The Web Single Sign-On (homogeneous application server) Runtime pattern is shown in Figure 4-11.

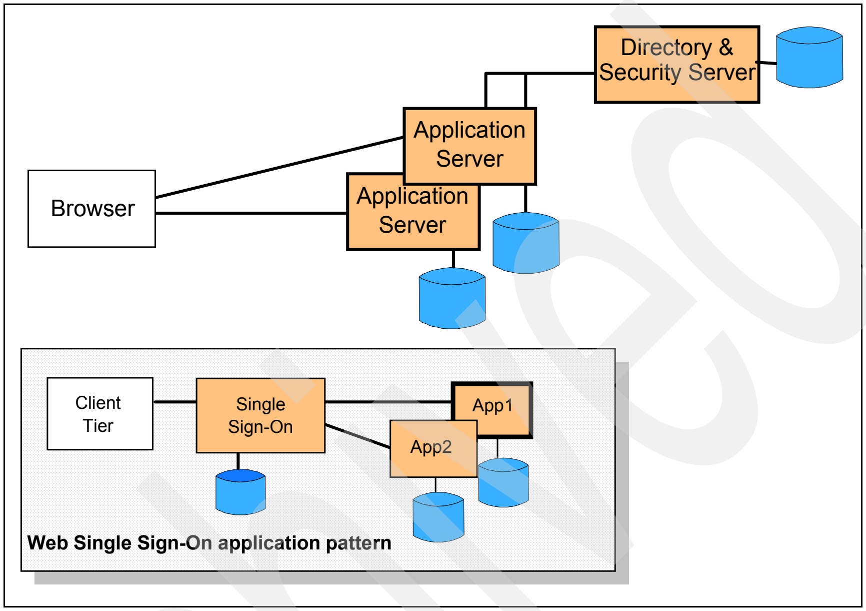


Figure 4-11 Single Sign-On (homogeneous application server) Runtime pattern

Runtime pattern - Access Integration: Personalized

There are three variations of this Runtime pattern, as follows:

- Participatory personalization

  Participatory personalization allows the user to design the content and the layout of the content that they see by explicitly choosing from a selection of options. The personalization service stores the customization results in a profile record.

- Predictive personalization

  Predictive personalization uses an inference engine to personalize content based on the past history (click stream) of user (all users) behavior. The inference engine may reference a user profile for demographic information to categorize users.

- Prescriptive personalization

  Prescriptive personalization provides a rules engine that delivers targeted content based on business rules defined by the enterprise. The personalization rules are based on data being requested by the user and the user's profile information.

Given the nature of the information the agents will be requesting (primarily work in progress) and to reduce the impact on their existing applications, ABC Insurance has indicated that it would like to provide a controlled and consistent look and feel to its agent community. Therefore, the personalization rules will be fixed based on a set of rules developed with the agent community and ABC Insurance. This requirement suggests the use of the Prescriptive personalized Runtime pattern for this solution. Figure 4-12 shows this Runtime pattern.

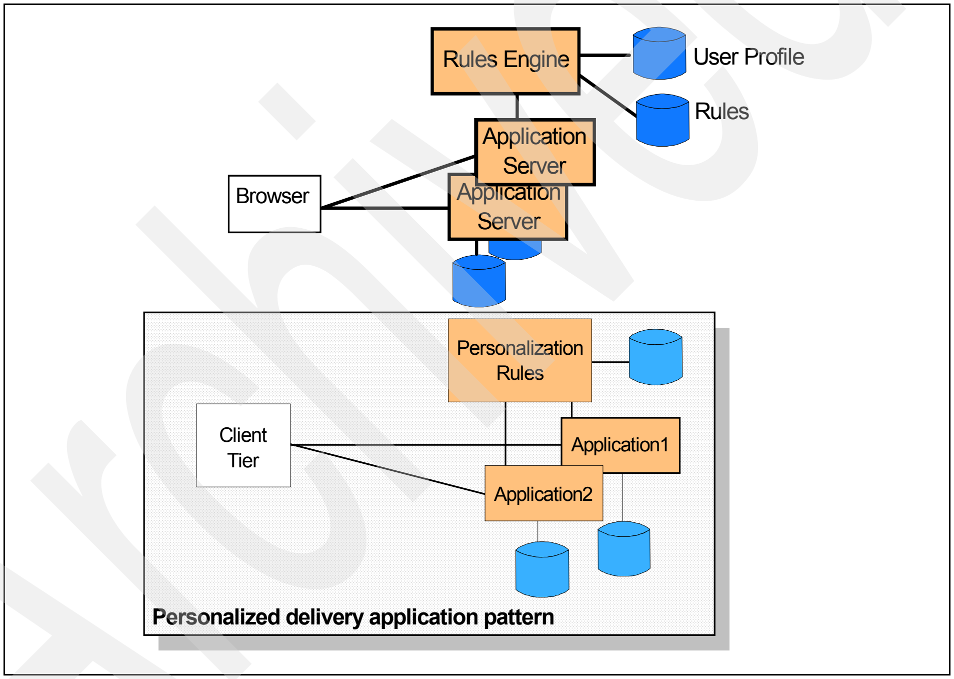


Figure 4-12 Prescriptive personalized Runtime pattern

In summary, based on the requirements provided by ABC Insurance and our analysis, we will be applying the following Runtime patterns to the ABC Insurance solution:

- Runtime pattern for the Router Application pattern
- Runtime pattern for the homogenous Web Single Sign-On Application pattern
- Runtime pattern for prescriptive Personalized Application pattern

A combined view of the Single Sign-On and Prescriptive personalized runtime pattern is shown on the Patterns Web site and in Figure 4-13.

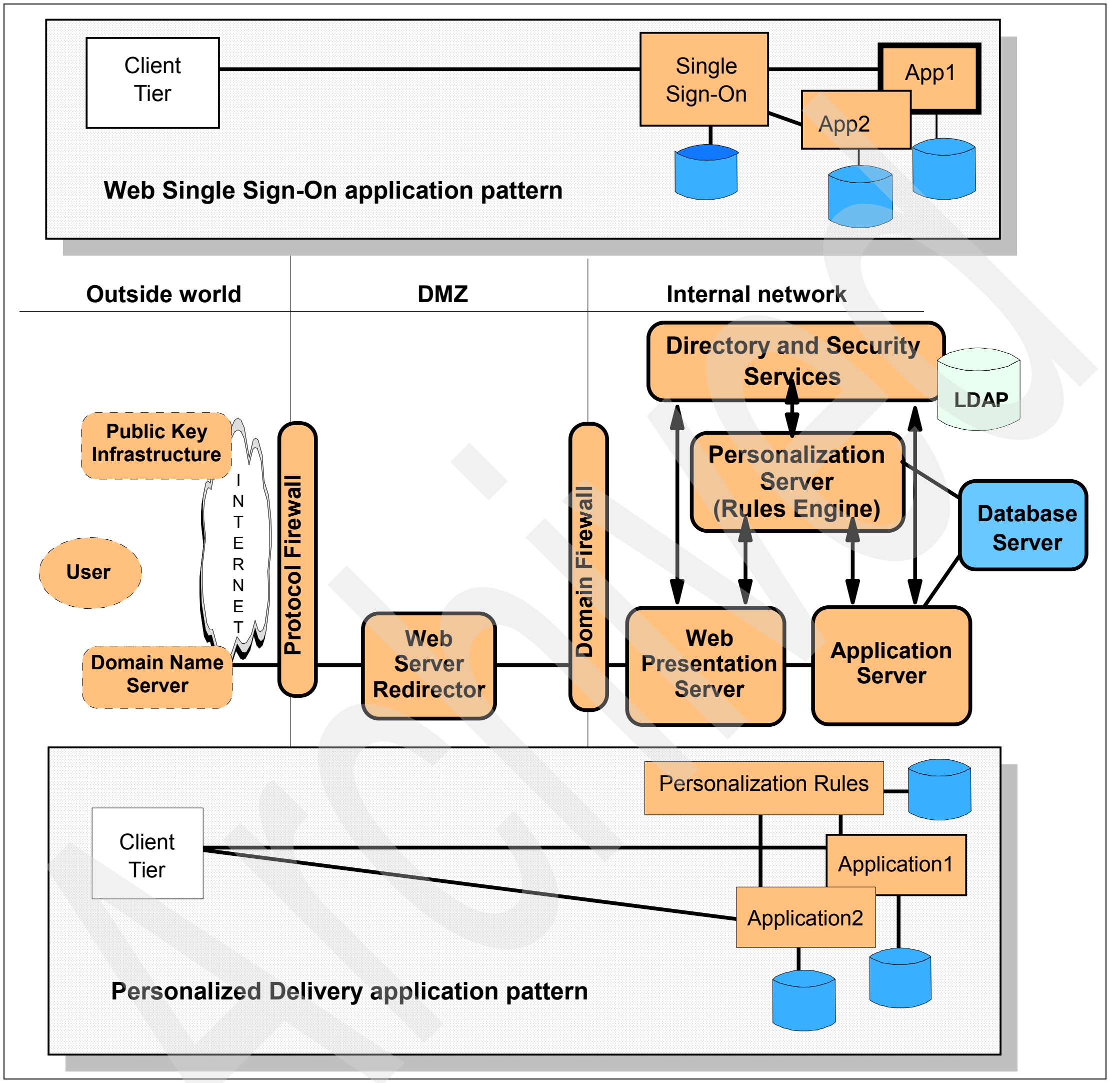


Figure 4-13 Combined Access Integration Runtime patterns

4.4.3 Macro Design phase

During the Macro Design phase, we begin to develop the complete specification of the architecture in line with the work carried out during the Solution Outline phase. The Macro Design phase also lays the groundwork in preparing for the subsequent releases by developing a robust user interface and architecture capable of withstanding reasonable business change through the delivery of these releases and beyond. The Patterns for e-business will further support the activities undertaken at this stage.

Identifying requirements

Whereas “Identifying requirements” during the Solution Outline phase referred to the activity of determining the key requirements influencing the overall architecture, we will now concentrate on refining the system specification based on all user requirements, business processes, and business requirements by elaborating the system requirements and specifications. This will enable us to:

- Make better estimates for the first release
- Provide an overall specification for the structure and topology of the system

This provides an opportunity to update the system context diagram based on any new or refined information obtained during this phase.

Defining the architecture

Many of the activities in this phase are similar to those carried out during the Solution Outline phase. In this phase, we will focus on specifying the overall architecture, both from a structural and topological point of view so that in a subsequent phase, a start can be made on its implementation. During this activity, confirmation will be sought of the decisions taken during the Solution Outline phase.

This is an opportunity to update the architecture overview diagram(s) based on any new information obtained thus far.

This effectively implies that we will verify during the Macro Design stage that any additional functionalities are catered for either by the products suggested by the Patterns for e-business catalog or that these functionalities will be delivered by other products or earmarked for custom development. Figure 4-14 provides an example of how the Patterns for e-business and Product mappings can be leveraged to deliver a product selection for each of the nodes in the runtime model.

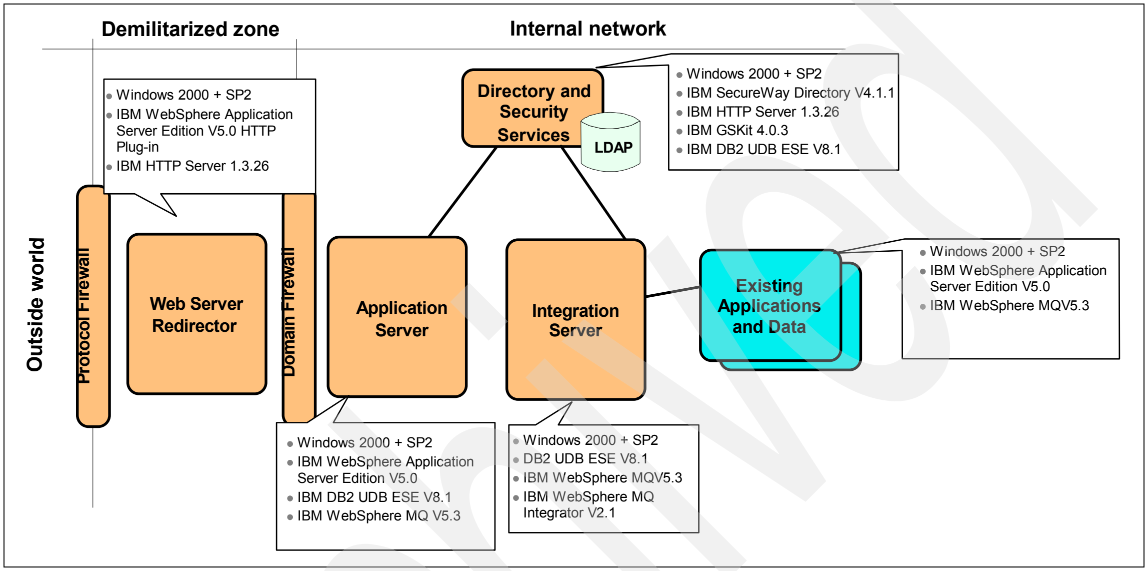


Figure 4-14 Product mappings for Self-Service: Router

Note: The source for all Product mapping information can be found on the Patterns for e-business Web site at the following URL:

http://www-106.ibm.com/developerworks/patterns/select-pattern.html

Building the development environment

The purpose of “Building the development environment” is to put the hardware and software into place to support the development activities and any associated tools.

The Patterns for e-business IBM Redbooks publications can also be leveraged in building this development environment. For each of the involved products, identified through the Product mappings, the architect should refer to the product installation instruction chapters in the related Patterns for e-business series IBM Redbooks publications.

4.4.4 Micro Design phase

The purpose of the Micro Design phase is to prepare for the build cycle(s) of a specific release of the system. This implies a further detailing of the results obtained in the prior two phases but narrowed down to the scope of the release itself. In this way, the foundation is established for all subsequent implementation related activities.

Refining the architecture

The main objective of the “Refining the architecture” activity during the Micro Design phase is to complete the design so that the build activities can be performed.

For design guidance related to a particular product, refer to the design and development chapters in Patterns for e-business series IBM Redbooks publications.

4.4.5 Build Cycle phase

The purpose of a Build Cycle phase is to incrementally develop and test part of the system until the objectives of the overall Release, within which the build cycle is embedded, are achieved.

Building the production environment

During this activity, we want to take a look again at the system's topology defined during the earlier phases and leverage this information to build the production environment. The Patterns for e-business series IBM Redbooks publications can further support building the production environment by referring to the product installation instruction chapters.

4.4.6 Deployment phase

The purpose of the phase is to deploy the system and prepare for the next release of the system.

Reviewing the solution plans

The purpose of this activity is to complete the systems management and software distribution plans.

This provides the basis for:

- Selecting system management tools.
- Selecting system management processes.

- Selecting and defining job descriptions, based on identified roles, to support the IT system.

For more information in relation to system management guidelines, refer to the system management guideline chapters in the Patterns for e-business series IBM Redbooks publications.

4.5 Summary

In this chapter, we have shown how to start from a business case and define a solution on the basis of multiple Business patterns. In the case study, a solution was defined, leveraging the Self-Service Business pattern and Access Integration pattern. We illustrated this for two different points in time, initially during proposal creation and later on during solution delivery. Each time, we demonstrated how the Patterns for e-business asset catalog was navigated and contributed significant value. The assets of the Patterns for e-business can deliver up to 80% of the proposed solution, and lend themselves very well to integration into an overall solution while at the same time making an important contribution to the overall quality of the solution.

4.5.1 Key terms and concepts

Refer to “Glossary” on page 203 for information about key terms and concepts.

4.6 Questions

Answer the following questions. To check your answers, see “Chapter 4: Answers to questions” on page 189.

4.6.1 Discussion and review questions

1. List the Proposal Development (Solution Design) activities identified in this chapter. Identify specifically what added value is provided by the IBM Patterns for e-business process.
2. This chapter illustrates Proposal Development activities in the context of the ABC Insurance case study. Given the findings in that case, please provide the following:
   a. What technical capabilities is ABC Insurance looking to implement?
   b. What advantages does ABC Insurance expect their customers to see with the implementation of this new technology?

   c. What business functions, specifically, will be provided by the solution?

3. List the steps followed to arrive at the recommended solution for ABC Insurance.
4. Which Business pattern(s) are applicable to the ABC Insurance case? Defend the pattern selection, specifically considering how it applies to the case.
5. Which Application pattern(s) are applicable to the ABC Insurance case? Defend the pattern selection, specifically considering how is applies to the case.
6. The Solution Delivery phase has several sub phases, including the Solution Outline phase and the Macro Design phase. In both sub phases, architecture definition occurs. Compare and contrast these two phases.
7. This chapter illustrates the Solution Startup phase with activities identified in the context of the ABC Insurance case study. Given the findings in that case, please provide the following:

   a. What is the focus of defining the project approach activity?

   b. Why would a systems analyst wish to become familiar with candidate assets during the activities of this phase?
8. What is the difference between a System Context Diagram, an Architecture Overview Diagram, and a Solution Overview Diagram?
9. In the context of the ABC Insurance case, which Business patterns are applicable? Why?
10. In the context of the ABC Insurance case, which Integration patterns are applicable? Why?
11. In the context of the ABC Insurance case, which Self-Service Application patterns are applicable? Why?
12. In the context of the ABC Insurance case, which Access Integration Application patterns are applicable? Why?
13. In the context of the ABC Insurance case, why was it decided to implement the homogenous application server approach?
14. In the context of the ABC Insurance case, how does the Identifying Requirements activity during the Solution Outline phase differ from the Macro Design phase?

4.7 Exercise and challenge questions

Complete the exercise and challenge questions for further understanding.

4.7.1 Challenge questions

1. This chapter provides a four-step procedure to arrive at the recommended solution, but the scope of the project limits consideration of factors that should, realistically, be incorporated when developing a budget for the project. Conduct research on an organization similar to ABC Insurance. Choose one of the factors identified under “Defining the solution” on page 88, (sizing estimates, available skills, performance) and expand on factors under that general category. What other factors are needed to realistically budget for the proposed solution that is not provided in the text? Make sure you can defend your answer with findings from research.
2. This chapter refers to Enterprise Application Integration (EAI) architecture, used when beginning the process of identifying applicable Application patterns. The chapter does not explicitly discuss an EAI-based architecture for ABC Insurance: we suggest that the solution under development is a component of ABC's overall EAI. Conduct research on the domain of EAI. What is it? Why is it significant for an organization to facilitate EAI in terms of e-business solutions?

4.7.2 Team case study projects

Here we discuss the team case study projects.

Proposal Development Activities

We will discuss several case studies in this section.

Case Study 1: NYPD changes the crime control equation by transforming the way it uses information

Read the Real-World Environment Case found at the beginning of Chapter 4: "NYPD changes the crime control equation by transforming the way it uses information."

Review 4.3, “Using the Patterns for e-business during solution delivery” on page 91 and 4.2.1, “Case study for proposal development” on page 87. Use these sections to guide your development of a project proposal for the NYPD Case.

Team assignment:

Based on the information communicated in the NYPD case, collaboratively analyze the following proposed development activities and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

1. Identify the requirements.
2. Document the client environment.
3. Define the solution.

Case Study 2: Bank of Montreal revitalizes IT infrastructure with flexible, scalable, and affordable technology with IBM

Read the case study “Bank of Montreal revitalizes IT infrastructure with flexible, scalable and affordable technology with IBM” that is available on the Web at:

http://www.redbooks.ibm.com/additional_materials/SG247466/Bank_of_Montreal.pdf

Review 4.3, “Using the Patterns for e-business during solution delivery” on page 91 and 4.2.1, “Case study for proposal development” on page 87. Use these sections to guide your development of a project proposal for the Bank of Montreal case.

Team assignment:

Based on the information communicated in the Bank of Montreal case, collaboratively analyze the following proposal development activities and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

1. Identify the requirements.
2. Document the client environment.
3. Define the solution.

Case Study 3:General Glass sees sales shine with best-of-breed IBM software for SMBs

Read the case study “General Glass sees sales shine with best-of-breed IBM software for SMBs” that is available on the Web at:

http://www.redbooks.ibm.com/additional_materials/SG247466/General_Glass.pdf

Review 4.2, “Using the Patterns for e-business during proposal development” on page 86 and 4.2.1, “Case study for proposal development” on page 87. Use these sections to guide your development of a project proposal for the General Glass case.

Team assignment:

Based on the information communicated in the General Glass case, collaboratively analyze the following proposal development activities and then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

1. Identify the requirements.
2. Document the client environment.
3. Define the solution.

Using Patterns during Solution Delivery

Here we discuss using Patterns during Solution Delivery.

Walk-through and Learn Case: ABC Insurance

Team assignment:

Collaboratively walk through this section and analyze the activities involved in the phases shown below. Then perform a project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

1. Solution Start-up phase
   a. Define the project approach.
2. Solution Outline phase
   a. Identify the requirements.
   b. Define the high level architecture.
3. Macro Design phase
   a. Identify the requirements.
   b. Define the architecture.
   c. Build the development environment.
4. Micro Design phase
   a. Refine the architecture.
5. Build cycle phase
   a. Build the production environment.
6. Deploy phase
   a. Review the solution plans.

Case Study 4:NYPD changes the crime control equation by transforming the way it uses information

Team assignment:

From the results of your team's analysis of the NYPD case, perform the phases for its solution delivery and then perform a project debriefing to gain insights and a better understanding of the phases and the application of the patterns approach.

1. Solution Start-up phase
   a. Define the project approach.
2. Solution Outline phase
   a. Identify the requirements.
   b. Define the high level architecture.
3. Macro Design phase
   a. Identify the requirements.
   b. Define the architecture.
   c. Build the development environment.
4. Micro Design phase
   a. Refine the architecture.
5. Build cycle phase
   a. Build the production environment.
6. Deploy phase
   a. Review the solution plans.

Case Study 5:Bank of Montreal revitalizes IT infrastructure with flexible, scalable, and affordable technology with IBM

Team assignment:

From the results of your team's analysis of the Bank of Montreal case, perform the phases for its solution delivery and then perform a project debriefing to gain insights and a better understanding of the phases and the application of the patterns approach.

1. Solution Start-up phase
   a. Define the project approach.

2. Solution Outline phase
   a. Identify the requirements.
   b. Define the high level architecture.
3. Macro Design phase
   a. Identify the requirements.
   b. Define the architecture.
   c. Build the development environment.
4. Micro Design phase
   a. Refine the architecture.
5. Build cycle phase
   a. Build the production environment.
6. Deploy phase
   a. Review the solution plans.

Case Study 6:General Glass sees sales shine with best-of-breed IBM software for SMBs

Team assignment:

From the results of your team's analysis of the General Glass case, perform the phases for its solution delivery and then perform a project debriefing to gain insights and a better understanding of the phases and the application of the patterns approach.

1. Solution Start-up phase
   a. Define the project approach.
2. Solution Outline phase
   a. Identify the requirements.
   b. Define the high level architecture.
3. Macro Design phase
   a. Identify the requirements.
   b. Define the architecture.
   c. Build the development environment.
4. Micro Design phase
   a. Refine the architecture.

5. Build cycle phase

   a. Build the production environment.

6. Deploy phase

   a. Review the solution plans.

4.7.3 Multiple choice

1. RFP stands for:

   a. Request for Provisions.

   b. Requirements for Processing.

   c. Request for Proposal.

   d. None of the above.

2. In the Proposal Development Activity of _____, the navigation of the Patterns for e-business asset catalog starts from a broad categorization of business requirements that guide asset selection and offer insights into the business area and the main business drivers.

   a. Identifying requirements.

   b. Documenting the client environment.

   c. Defining the solution.

   d. None of the above.

3. In the Proposal Development Activity of _____, the information found in the RFP will help document the current IT environment, which is key during the later stages of the asset selection process and at the same time creates an opportunity for a focused analysis of available documentation.

   a. Identifying requirements.

   b. Documenting the client environment.

   c. Defining the solution.

   d. None of the above.

4. Which key element was contributed by the Patterns for e-business process to the ABC Insurance case?

   a. Identification of required functionalities.

   b. Product selection guidance.

   c. Information regarding the interactions between products.

   d. All of the above.

5. The Business pattern _____ is applicable to the ABC Insurance case due to the requirement for supporting the agents through a single, consistent, and seamless access mechanism to the various applications.

   a. Self-Service.

   b. Access Integration.

   c. Both A and B are correct.

   d. Neither A or B is correct.

6. The Application pattern _____ is identified in part by the need for ABC Insurance agents to access the multiple existing systems across a Web channel through a single sign-on interface.

   a. Self-Service::Router Application pattern.

   b. Access Integration::Web Single Sign-on Application pattern.

   c. Both A and B are correct.

   d. Neither A or B is correct.

7. Based on the information obtained from the Patterns for e-business, ABC Insurance has the opportunity to expand the proposal by:

   a. Estimating the man days needed to build new functionality or to integrate existing functionality.

   b. Estimating the software budget, taking into account, among other aspects, the suggested software products.

   c. Both A and B are correct.

   d. Neither A or B is correct.

8. In the Solution Delivery Phase of _____, the proposed approach is confirmed and refined based on additional information garnered. This is usually done with the project team and client personnel.

   a. Solution Startup.

   b. Solution Outline.

   c. Macro Design.

   d. None of the above.

9. In the Solution Delivery Phase of _____, the focus of this activity is to gather overall business requirements and objectives concentrating on those that will have a significant impact on the solution.

   a. Solution Startup.

   b. Solution Outline.

   c. Macro Design.

   d. None of the above.

10. The Business pattern provides richness to the System Context Diagram by:

   a. Relating it to similar trends in other industries.

   b. Allowing for the validation of the selected patterns.

   c. Both A and B are correct.

   d. Neither A or B is correct.

11. The main objective of this activity is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be given to more challenging areas of the proposed solution.

   a. Solution Outline sub-phase: Defining the high level architecture activity.

   b. Solution Outline sub-phase: Identifying requirements activity.

   c. Macro Design sub-phase: Defining the architecture activity.

   d. Macro Design sub-phase: Identifying requirements activity.

12. The architecture is defined in the ___ and then confirmed and refined in the ___.

   a. Solution Outline sub-phase: Macro Design sub-phase.

   b. Macro Design sub-phase: Solution Outline sub-phase.

   c. Solution Design sub-phase: Micro Design sub-phase.

   d. None of the above.

13. During the ______ , the emphasis lies much more on identifying all required functionalities than on exploring each one in detail. The overall objective is to get an accurate view of the overall effort required to deliver the system.

   a. Solution Startup phase.

   b. Solution Outline phase.

   c. Macro Design phase.

   d. Micro Design phase.

14. In terms of actors, roles, and users:

    a. Several users can play the same role.

    b. One user can perform several roles.

    c. Both A and B are correct.

    d. Neither A or B is correct.

15. The _____ diagram gives an overview of the system that will be delivered: it shows the building blocks of the system itself.

    a. Architecture Overview.

    b. Solution Overview.

    c. Both A and B are correct.

    d. Neither A or B is correct.

16. The actors and functions identified in the high-level business description are transformed into a visual representation in the _____ diagram.

    a. System Context.

    b. Solution Overview.

    c. Both A and B are correct.

    d. Neither A or B is correct.

17. On the Solution Overview diagram, existing or older systems are represented with:

    a. A box.

    b. Double lines on a box.

    c. A shaded box.

    d. None of the above.

18. On the Solution Overview diagram, business functions must link up with:

    a. Another business function.

    b. An actor.

    c. Both A and B are correct.

    d. Neither A or B is correct.

19. The ______ diagram provides the initial structure of a system on which to overlay the identified Business and Integration patterns.

    a. System Context.

    b. Solution Overview.

    c. Both A and B are correct.

    d. Neither A or B is correct.

20. In the ABC Insurance case, the processing of electronic payments dictates the need to apply the _____ Business pattern, which accommodates the needed external payment processing center.

    a. Collaboration.

    b. Information Aggregation.

    c. Extended Enterprise.

    d. None of the above.

21. In the ABC Insurance case, the _____ Business pattern fits, given that the agent is a user of the ABC Insurance business systems and ABC Insurance's wish to create a personalized experience to the agents.

    a. Collaboration.

    b. Information Aggregation.

    c. Extended Enterprise.

    d. None of the above.

22. In the ABC Insurance case, the _____ Integration pattern fits, given need for a common portal to provide front-end integration of multiple services and information.

    a. Access Integration.

    b. Application Integration.

    c. Portal Integration.

    d. None of the above.

23. In the ABC Insurance case, the _____ Integration pattern fits, given the need to provide a seamless back-end integration of multiple applications and data without accessing them directly.

    a. Access Integration.

    b. Application Integration.

    c. Portal Integration.

    d. All of the above.

24. In the ABC Insurance case, the _____ Self-Service Application pattern supports the capability of processing insurance applications across a single line of business.

   a. Router.

   b. Decomposition.

   c. Agent.

   d. None of the above.

25. In the ABC Insurance case, the _____ Access Integration Application pattern provides a framework for giving access to applications and information tailored to the interests and roles of a specific user, such as an agent.

   a. Pervasive Device Access.

   b. Extended Single Sign-on.

   c. Personalized Delivery.

   d. None of the above.

26. In the _____ environment of a Runtime pattern for Web Single Sign-On, all applications can exploit the application's security server for single sign-on functionality.

   a. Homogenous.

   b. Heterogeneous.

   c. Both A and B are correct.

   d. Neither A or B is correct.

27. The ____ variation of the Runtime pattern allows the user to design the content and the layout of the content that they see by explicitly choosing from a selection of options.

   a. Participatory personalization.

   b. Predictive personalization.

   c. Prescriptive personalization.

   d. All of the above.

28. During the _____ , the groundwork is laid for preparing for subsequent releases by developing a robust user interface and architecture capable of withstanding reasonable business change through the delivery of these releases and beyond.

    a. Solution Startup phase.

    b. Solution Outline phase.

    c. Macro Design phase.

    d. Micro Design phase.

29. During the ______ , the verification of any additional functionalities are catered for either by the products suggested by the Patterns for e-business catalog or because these functionalities will be delivered by other products or earmarked for custom development.

    a. Solution Startup phase.

    b. Solution Outline phase.

    c. Macro Design phase.

    d. Micro Design phase.

30. The purpose of the ______ is to prepare for the build cycle(s) of a specific release of the system.

    a. Solution Startup phase.

    b. Solution Outline phase.

    c. Macro Design phase.

    d. Micro Design phase.

31. The purpose of the ______ is to deploy the system and prepare for the next release of the system.

    a. Solution Startup phase.

    b. Solution Outline phase.

    c. Macro Design phase.

    d. Deployment phase.

4.7.4 Fill in the blank

1. The information in the ________________ is key during the asset selection process, as it creates an opportunity for a focused analysis of the document.

2. Patterns for e-business contribute significantly to the ________________ of the solution to be proposed.

3. In a client environment that requires delivery of multiple applications, each with its own authentication procedure, you can simplify access through enabling ________________________ within the infrastructure.

4. The results of the selection process, expressed in terms of required software products, can be used for the __________________ aspect of the solution design.

5. The _______________________ represents the proposed solution emphasizing the interactions to be supported with its environment, both in terms of people or systems.

6. The main objective of the micro design phase is to complete the design so that the ____________ activities can be performed.

7. A(n) ____________ represents a coherent set of roles.

8. In the Router Application pattern, the ____________ tier serves as an integration point for delivery channels in the presentation tier, allowing access to multiple back-end applications.

9. Prescriptive personalization provides a rules engine that delivers targeted content based on ___________________ defined by the enterprise.

4.7.5 True or false

1. In the Proposal Development activity known as Defining the Solution, the navigation of the Patterns for e-business asset catalog starts from a broad categorization of business requirements that guide asset selection and offer insights into the business area and the main business drivers.

2. In the Proposal Development Activity known as Documenting the Client Environment, the information found in the RFP will help document the current IT environment, which is key during the later stages of the asset selection process and at the same time creates an opportunity for a focused analysis of available documentation.

3. The Application pattern focuses on the application level functionality required to support the identified Business pattern(s).

4. Patterns for e-business should be view as a replacement for traditional Requirements Determination and Design methods.

5. The Business pattern provides richness to the System Context Diagram by relating it to similar trends in other industries while at the same time allowing for the validation of the selected patterns.

6. The focus of the Building the Production Environment activity is to build the production environment based on the architecture defined in previous phases of the Solution Delivery process.

7. The focus of the Solution Startup phase known as Defining the Project Approach lies in familiarizing the project team with the development method, assessing training, and mentoring needs, and tailoring the development method to the needs of the project, taking the use of assets into account.
8. A business function represents a high-level specification extracted from the low-level business requirements.
9. The actors and business functions obtained from the high-level business description will serve as the basis for developing several deliverables during the Solution Outline phase.
10. Architecture Overview diagrams are never produced for different target audiences.
11. Business and IT drivers are a consistent set of objects that assist the architect in selecting the appropriate Business and Integration Patterns.
12. There are two variations of the Runtime pattern for Web Single Sign-On.
13. The purpose of building the development environment is to put the hardware and software into place to support the development activities and any associated tools.
14. The purpose of the Macro Design phase is to prepare for the build cycle(s) of a specific release of the system.

4.7.6 Short answer

1. List the Proposal Development (Solution Design) activities identified in this chapter.
2. This chapter illustrates Proposal Development activities in the context of the ABC Insurance case study. Given the findings in that case, what business functions will be provided specifically by the solution?
3. Which Business pattern(s) are applicable to the ABC Insurance case? Defend the pattern selection as to how it applies specifically to the case.
4. Which Application pattern(s) are applicable to the ABC Insurance case? Defend the pattern selection as to how it applies specifically to the case.
5. This chapter illustrates the Solution Startup phase with activities identified in the context of the ABC Insurance case study. Given the findings in that case, why would a systems analyst wish to become familiar with candidate assets during the activities of this phase?
6. What is the difference between a System Context Diagram, an Architecture Overview Diagram, and a Solution Overview Diagram?

5

Applying Patterns with existing Custom designs

Sometimes a middleware requirement identifies itself; for example, there could be a need to integrate WebSphere and Domino and WebSphere and SAP. This chapter focuses on these technical solutions and some examples of useful assets that we describe as existing Custom designs.

Outline

- Introduction to Existing Custom designs
  - WebSphere and Domino integration
    - Business and IT drivers
    - Selecting the patterns
    - Hybrid Runtime patterns

      Navigating the hybrid Runtime patterns
  - WebSphere and SAP integration
    - Selecting the patterns

      Self-Service designs for small and mid-sized manufacturing companies

      Application integration and extended enterprises designs for large manufacturing companies

    Electronic commerce for large distributor/retailer enterprises

  - WebSphere and Edge Server integration
    - High availability principles

      Runtime patterns for high availability
    - Product mapping

      Product mapping option 1

      Product mapping option 2
    - Performance factors

      Runtime patterns for performance
- Summary

Learning objectives

The objectives of this chapter help you to:

- Understand and explain a solution built on an Existing Custom Design Pattern.
- Understand and explain the value added by IBM Patterns for e-business.
- Explain the reasoning behind the chosen solution.
- Explain the interactions between the chosen patterns.
- Explain the parts of the chosen solution and how those parts interact.

Real-world environment

In this section, we refer you to a real-world situation that uses the concepts in this chapter. The case study “Volkswagen's world-class procurement strategy produces breakthrough productivity gains” is available on the Web at:.

http://www.redbooks.ibm.com/additional_materials/SG247466/Volkswagen_World-class_procurement_strategy.pdf

5.1 Introduction to existing Custom designs

The Patterns for e-business are designed to address the majority of e-business applications an enterprise might need or want to develop. Because most enterprises have similar e-business needs, successful solutions can often be reused. Business, Integration, and Composite patterns provide a certain amount of flexibility in creating varied e-business applications to address minor differences in enterprise needs.

At the time of the writing of this IBM Redbooks publication, Patterns for e-business had identified three existing Custom designs that could be exploited as reusable assets. The three existing Custom designs are as follows:

1. WebSphere and Domino Integration
2. WebSphere and SAP Integration
3. WebSphere and Edge Server Integration

Now, let us look more closely at the individual existing Custom designs and how this type of technical solution may be encountered.

5.1.1 WebSphere and Domino integration

Certain circumstances will, almost automatically, lead to addressing the integration of Domino and WebSphere in a solution under some fairly well understood circumstances.

WebSphere and Domino Integration solutions combine the transactional functionality of the Self-Service Business pattern with the collaborative functionality of the Collaboration Business pattern. The Self-Service Business pattern is commonly observed in e-business solutions that provide users the ability to access their information and change it by interacting directly with core business systems and databases. This pattern captures the essence of direct interactions between users and the enterprise. Such interactions can range from simple static information lookup to complex updates involving enterprise data. The Collaboration Business pattern occurs in e-business solutions that involve one or 2-way interactions between users of the solution.

Examples of applications that use the Self-Service Business pattern in combination with the Collaboration Business pattern include the following:

- Customer-facing applications, such as an Online Broker application, that allow investors to manage their portfolios and make equity trades across the Web, and also allow investors to collaborate with the broker's help desk
- Business partner-facing applications, such as a dealer extranet, that allow dealers of a manufacturing company to link their sales distribution operations to the shop floor processes by exchanging forecast, demand, and production information, and also facilitate online collaboration with the manufacturer's sales department

To validate the use of these patterns and the integration between them, we must take a look at the business and IT drivers. These are a consistent set of objectives that assist the architect in selecting the appropriate Business and Integration patterns.

Business and IT drivers

Businesses developing a solution and needing the following characteristics should consider using a WebSphere/Domino integration design:

- The users and customers need to directly interact with business processes.
- The business process needs to be integrated with existing business systems and information.
- The business process must be reachable in a common, consistent, and simplified manner through multiple delivery channels.
- The business activity demands and fosters collaboration and the sharing of information among its participants.

Selecting the Patterns

The Patterns approach for integrating WebSphere and Domino is based on a Custom design that involves the following Patterns shown in Figure 5-1.

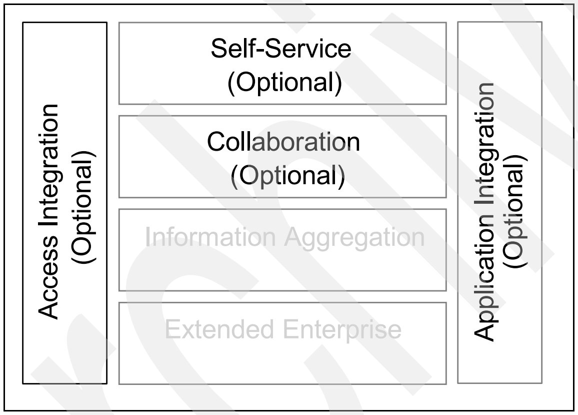


Figure 5-1 WebSphere/Domino Integration as a Custom design

As shown in Figure 5-1, the WebSphere/Domino Integration Custom design represents multiple different combinations of the Self-Service and Collaboration business patterns, along with the Access and Application Integration patterns. These solutions take advantage of the transactional and collaborative powers of the WebSphere and Domino product lines and address the issues that arise when integrating these servers together in a common e-business application.

To validate that the business problem can be solved by a WebSphere/Domino Custom design, refer to the WebSphere/Domino Integration general design guidelines on the Patterns Web site or to *Applying the Patterns for e-business to Domino and WebSphere Scenarios*, SG24-6255. This IBM Redbooks publication is now updated with the newest Domino and WebSphere releases; refer to *Patterns: Custom Designs for Domino and WebSphere Integration*, SG24-6903.

If it has been determined that the WebSphere/Domino Custom design might provide an appropriate solution design for your business need, you should review the various methods to connect WebSphere and Domino to one another at the Patterns for e-business Web site:

http://www-106.ibm.com/developerworks/patterns/ws-domino/index.html

Hybrid Runtime patterns

If you already understand the issues involved when selecting an appropriate connection method for your solution, the next step is to select a hybrid Runtime pattern. The WebSphere/Domino Integration design set includes numerous hybrid Runtime designs, providing solution flexibility to address the specific needs of the business process being automated.

The specific business functionality supported by applications that integrate WebSphere and Domino varies from one industry to another, and for each individual implementation of this type of solution. Patterns architects have performed a somewhat improvised assessment of the various solutions that might be created which combine WebSphere and Domino, and anticipated the most common problems developers would encounter when building these applications, along with solutions that can be used to address these problems.

There are two sets of hybrid Runtime patterns that are WebSphere and Domino release-dependent. This example uses the older hybrid Runtime patterns. For the newest Runtime patterns, refer to *Patterns: Custom Designs for Domino and WebSphere Integration*, SG24-6903.

Navigating the hybrid Runtime patterns

There are two methods you can use to navigate through the hybrid Runtime patterns. They are as follows:

- The first is simply to move through all the Runtime patterns in the linear order in which they are presented on the Patterns for e-business Web site. This method provides an in-depth review of all the WebSphere/Domino Integration designs.
- The second is to use the matrix shown in Table 5-1 on page 142.

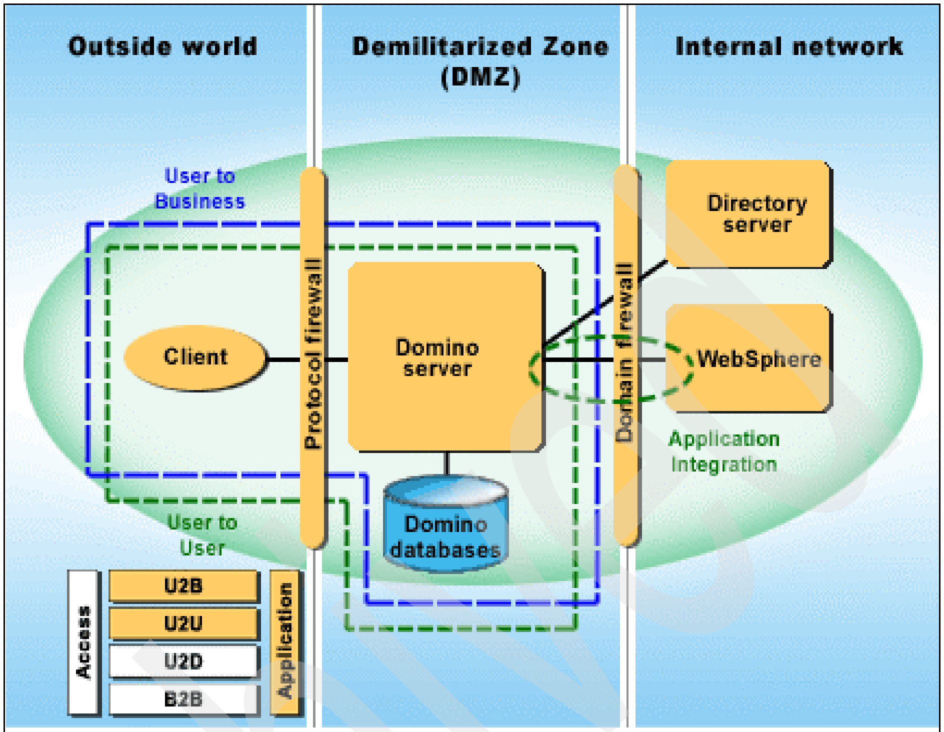
  If you already have an understanding of the required functionality, complexity, and extensibility of the solution you are designing, using Table 5-1 on page 142 is recommended. It divides the WebSphere/Domino Integration designs into five categories based on focus of functionality. It then further divides the designs according to the complexity and extensibility of each.

The numbering system used to represent each solution design in Table 5-1 is based upon the chapter and section number where the design is found in the IBM Redbooks publication *Applying the Patterns for e-business to Domino and WebSphere Scenarios*, SG24-6255.

Table 5-1 Navigating the hybrid Runtime patterns

Pattern	Entry-level	More advanced	More extensible
Collaboration-Centric	3.1.2, 3.1.3	3.1.5	3.1.12, 3.1.14, 8.8
Transaction-Centric	3.1.4	3.1.5	3.1.13
Dual Focus	3.1.5	8.1.2.1	3.1.6, 3.1.13, 8.1.2.1
Collaboration Only	3.1.7, 8.9	3.1.10	3.1.11, 8.3
Transaction Only		Self-Service	Business pattern

For example, a collaboration-centric hybrid Runtime pattern is captured in Figure 5-2 on page 143.



©Copyright IBM Corporation, 2001. All rights reserved.

Figure 5-2 Domino application with WebSphere Integration platform (hybrid Runtime 3.1.2)

The designs are referred to as "hybrid Runtime patterns" because they show both the middleware nodes commonly seen on a normal Runtime pattern and a graphical representation of the Business and Integration patterns used in combination with one another to create this solution design.

Additionally, because these hybrid Runtime patterns are directly constructed with WebSphere and Domino in mind, the middleware node names are mapped directly to Domino, WebSphere, and related products in the figures, instead of generic node names.

Additional identified hybrid Runtime patterns are as follows:

- WebSphere loaded on a Domino Server (hybrid Runtime pattern 3.1.3)
- WebSphere applications with Domino services (hybrid Runtime pattern 3.1.4)
- Single sign-on between Domino and WebSphere (hybrid Runtime pattern 3.1.5)

- Using a security server to manage authentication and connections (hybrid Runtime pattern 3.1.6)
- Domino User-to-User (hybrid Runtime pattern 3.1.7)
- Sametime® User-to-User (hybrid Runtime pattern 3.1.8)
- Combined Sametime and Domino (single machine) (hybrid Runtime pattern 3.1.9)
- Combined Sametime and Domino (separate machine) (hybrid Runtime pattern 3.1.10)
- Sametime and Domino with Policy Director (hybrid Runtime pattern 3.1.11)
- Single sign-on with Sametime, Domino, and WebSphere (hybrid Runtime pattern 3.1.12)
- Combined Sametime, Domino, and WebSphere with Policy Director (hybrid Runtime pattern 3.1.13)
- Domino as a central loosely-coupled broker (hybrid Runtime pattern 3.1.14)
- Combining caching proxy and load balancer (hybrid Runtime pattern 8.1.2.1)
- Using Tivoli® Policy Director as a load balancer (hybrid Runtime pattern 8.3)

For more details on WebSphere and Domino Integration Custom designs, refer to the Patterns for e-business Web site at the following URL:

http://www-106.ibm.com/developerworks/patterns/ws-domino/index.html

and the IBM Redbooks publication *Applying the Patterns for e-business to Domino and WebSphere Scenarios*, SG24-6255.

5.1.2 WebSphere and SAP integration

The Patterns for e-business designs can be implemented using any software component that fulfills the runtime requirements of the solution, whether those components are IBM products or from a third-party vendor. The specific business functionality supported by applications that integrate WebSphere and SAP varies from one industry to another, and for each individual implementation of this type of solution. The designs that follow illustrate solutions that address Self-Service, Extended Enterprise, Application Integration, and Electronic Commerce functionality for use by small to mid-sized business, large enterprises, and large enterprise distributor/retailers, respectively.

Selecting the Patterns

Several models have been developed to assist the architect in selecting the correct Runtime patterns for the WebSphere and SAP integration. These models are broken down into the following types of enterprises:

- Self-Service designs for small or mid-sized manufacturing companies
- Application Integration and Extended Enterprises designs for large manufacturing enterprises
- Electronic Commerce for large distributor/retailer enterprises

We will now take a look at each of these development scenarios in further detail.

Self-Service designs for small and mid-sized manufacturing companies

In Figure 5-3, the WebSphere/SAP integration Custom design makes use of SAP products in combination with the Self-Service business pattern and the Application Integration pattern for small or mid-sized businesses.

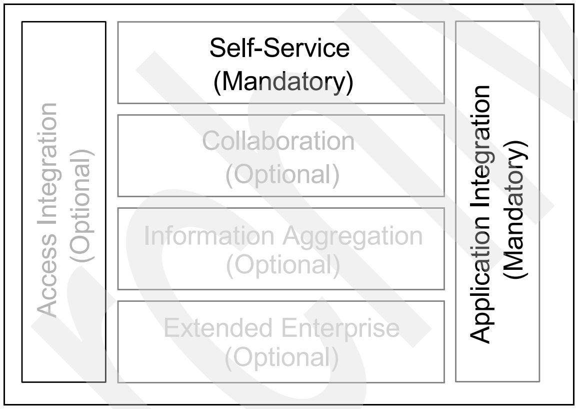


Figure 5-3 Self-Service designs for small or mid-sized manufacturing companies

The following criteria are considerations for companies fitting this model:

- The company has finished successful implementation of SAP R/3®.
- The company has challenges in running the standard SAP R/3.
- The integration between the outside production and payroll system is done through periodic file transfer and batch input to SAP R/3.
- The company has not planned for a transition to mySAP™.com because of complexity and licence cost issues.
- The company wants to open up for the internet and e-business.

Companies in small and medium businesses often use SAP standard R/3 with the SAP GUI interface and often do not have mySAP.com® components installed.

For more information about using SAP products to create Self-Service applications for small to mid-sized manufacturing companies, please review pages 3 through 10 of the PDF presentation located on the Patterns for e-business Web site. Refer to the following URL:

http://www-106.ibm.com/developerworks/patterns/sap/select-runtime-topology.html

Application integration and extended enterprises designs for large manufacturing enterprises

This WebSphere/SAP Integration Custom design represents multiple different combinations of the following patterns for use by large manufacturing enterprises that are similar to the example criteria for use in Figure 5-4.

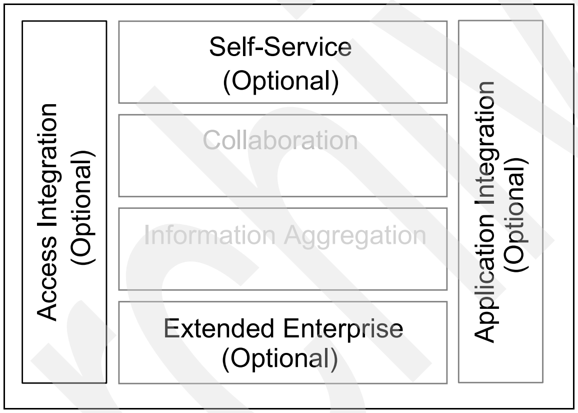


Figure 5-4 Application Integration and Extended Enterprises designs for large manufacturing enterprises

An example of applying these Patterns follows:

A holding company wants to have an SAP roll out over all subsidiaries and plans to do this over five years. They have a mix of 390, Unisys, and NT applications, but the main existing systems run on 390. To reduce risk, the company plans to roll out SAP in three waves: first, SAP Finance, then SAP Logistics, and third, mySAP.com SCM. In addition, they plan to implement the automotive industry solution.

The company made a detailed analysis of which functions can be migrated to SAP R/3 and found that in total 60% of their business could be handled with SAP

R/3. 40% of their business, which consists mainly of the HR payroll system, PPS, CAD, a 390 procurement system and some PC applications, remain outside of the functionality range of SAP.

Additional requirements are to have a common middleware/integration technology to support internal and internet access needs.

The company today does not have ABAP™ or Java skills but wishes to develop both. They recognize that most of their younger employees are more interested in learning Java.

For more information about using SAP products to create Application Integration and Extended Enterprises applications for large manufacturing enterprises, please review pages 11 through 26 of the PDF presentation located on the Patterns for e-business Web site. Refer to the following URL:

`http://www-106.ibm.com/developerworks/patterns/sap/select-runtime-topology.html`

Electronic commerce for large distributor/retailer enterprises

This WebSphere/SAP Integration Custom design represents multiple different combinations of the following patterns for use by large distributors and retailers. as shown in Figure 5-5.

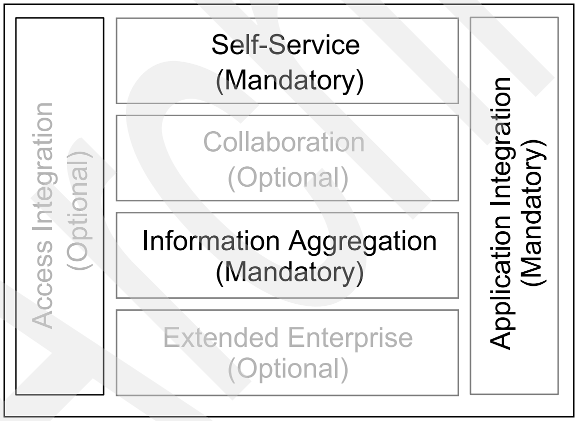


Figure 5-5 Electronic Commerce large distributor/retailer enterprises

After successful implementation of SAP R/3, the company wants to implement an internet front end that shows the latest product information and some CRM functions with the following tasks:

- Share shipping and product availability information with customers.
- Share product prices where special customer arrangements apply.
- Leverage field sales and service access using pervasive services.

- Provide real-time order tracking for customers.

In addition, they would like to build up a Web shop for customers. This should include the following:

- A personalized catalog
- Order entry to SAP R/3
- Price, shipping, and availability information (same as for CRM)
- Order tracking (same as for CRM)
- Payment through credit card and automatic update of SAP Finance accounts

Since this is essential for the company, it must be available twenty-four hours a day and orders must be accepted regardless of whether the SAP system is available. Orders must not get lost under any circumstances.

For more information about using SAP products to create Electronic Commerce applications for large distributors/retailers, please review pages 27 through 31 of the PDF presentation, located on the Patterns for e-business Web site. Refer to the following URL:

`http://www-106.ibm.com/developerworks/patterns/sap/select-runtime-topology.html`

5.1.3 WebSphere and Edge Server integration

Many companies will need to address the integration of high availability and performance into their e-business solutions. This existing Custom design focuses on Runtime patterns that address the functionality needed to combine WebSphere with High Availability and performance, which leads to WebSphere and Edge Server integration.

Runtime patterns:

- Describe the logical architecture required to implement an Application pattern.
- Depict the major middleware nodes, their roles, and the interfaces between the nodes.
- Address the processing logic and data placed on the middleware nodes.
- Can be represented by a logical node diagram.

Runtime patterns demonstrate Service level characteristics, specifically Non-Functional requirements such as:

- Availability
- Performance
- Scalability

- Security

The focus of this existing Custom design is the available Runtime patterns and their associated Product mappings.

The Runtime patterns are divided into two categories:

- Runtime patterns for high availability
- Runtime patterns for performance

The concentration is on high availability and high performance guidelines and scenarios as they apply to nodes in the Demilitarized Zone (DMZ) for Runtime patterns and Product mappings. Let us look at some examples of these Runtime and Product mappings.

There are two sets of Product mappings, depending on whether WebSphere V4 or WebSphere V5 is being used. The following descriptions are based on the V4 WebSphere products. For the V5 WebSphere products, refer to the Patterns for e-business Web site at:

http://www.ibm.com/developerworks/patterns

High availability principles

The basic principles of high availability (HA) are very simple. You must consider the possible failure of each component in the architecture of your solution. This will have an impact on your decisions all the way down from the architecture to the design, technology, topology, and products used to build the overall solution.

High availability rules for Runtime environments are as follows:

- Every entity must be redundant.
- Each entity must be monitored for failure.
- A failed entity must not receive any work.

Runtime patterns for high availability

Runtime patterns are shown in graphical form using standard node types to identify the function represented by that node. Most patterns will consist of a core set of common nodes, with the addition of one or more nodes unique to that Runtime pattern.

The Runtime patterns for high availability start off with a Base Runtime pattern for the Self-Service::Stand-Alone Single Channel Application pattern. This pattern is used as the standard Application pattern for the high availability patterns and variations in order to show the additional nodes you will need for each high availability configuration. From this Base Runtime pattern, four variations for high availability have been identified. For example, Figure 5-6 shows the Basic Runtime pattern variation 2: Load balancer hot standby.

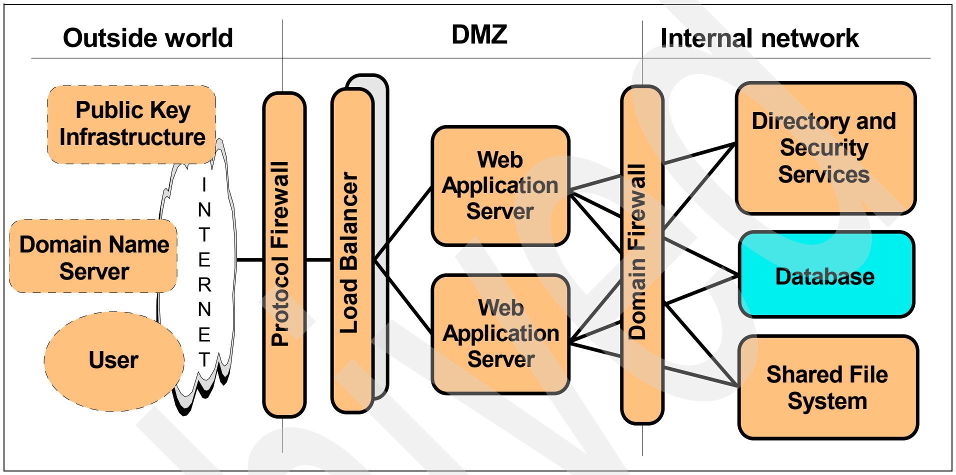


Figure 5-6 Basic Runtime pattern variation 2: Load balancer hot standby

This is the standard high availability solution for the load balancer nodes themselves. This Runtime pattern removes the single point of failure from the load balancer node by adding a second node with the same function. The first node will be an active, working load balancer. It is the primary load balancer. The second one keeps track of the balancing information and is ready to take over the work of the primary node at any moment. It is the standby load balancer.

Benefits include the following:

- Very simple installation and configuration.
- Very short delay for the forwarding task if the MAC forwarding method is used.
- High availability and failover for the Web application server nodes.
- Horizontal scalability is possible by simply adding more servers to the content server load balancer cluster.
- High availability and failover provided for the load balancer nodes.

Limitations include:

- One server in hot standby mode, not doing any productive work

Product mapping

There are two Product mapping options available for the load balancer hot standby configuration.

Product mapping option 1

In this Product mapping, shown in Figure 5-7, two separate load balancer machines are used to perform the primary and standby load balancer functions.

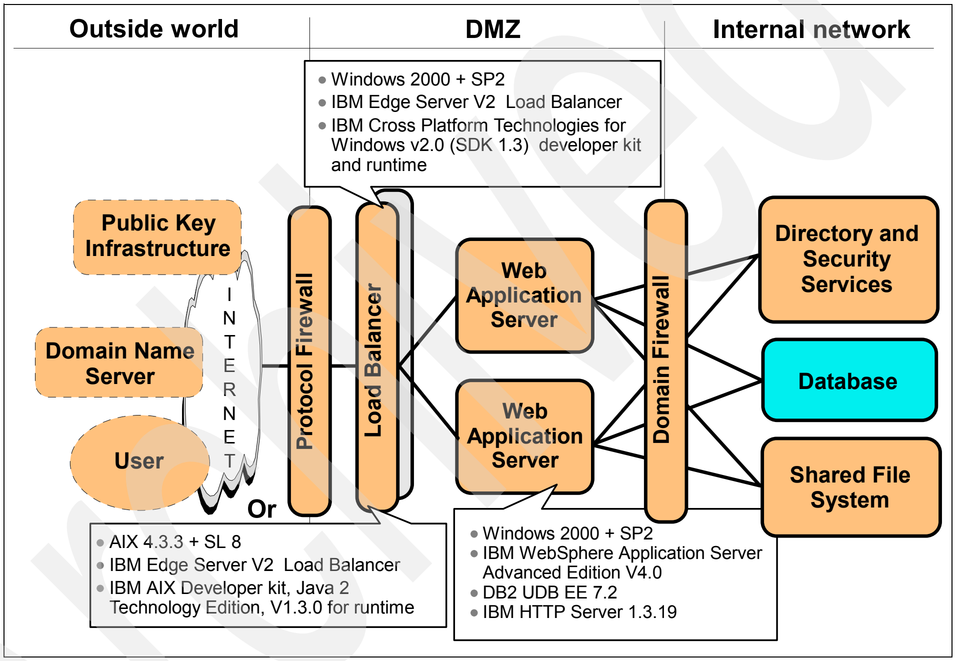
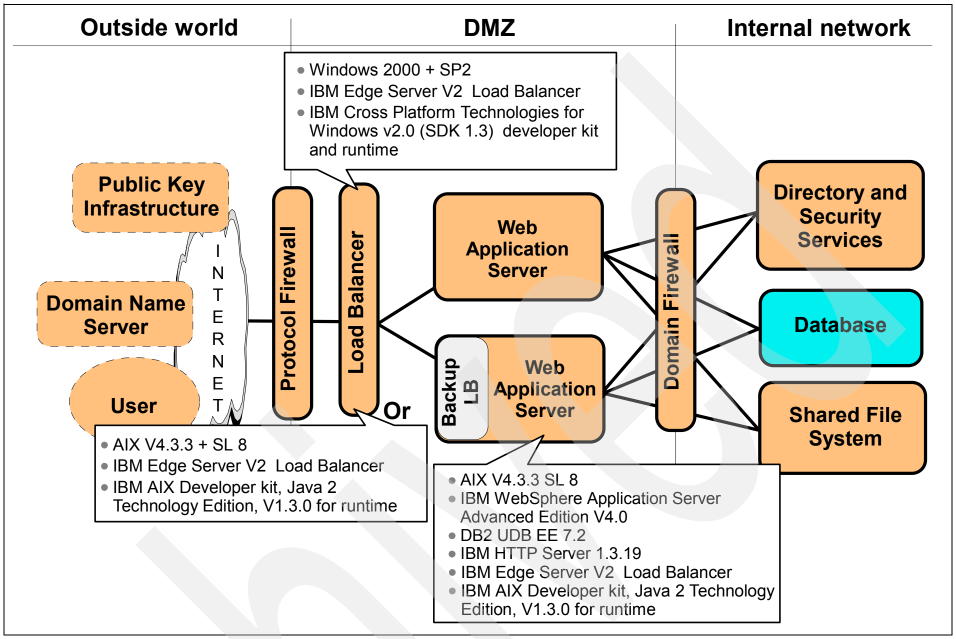

Outside world
DMZ
Internal network
Windows 2000 + SP2
IBM Edge Server V2 Load Balancer
IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3) developer kit and runtime
Public Key Infrastructure
Domain Name Server
User
INTERNET
Or
Protocol Firewall
Load Balancer
Web Application Server
Web Application Server
Domain Firewall
Directory and Security Services
Database
Shared File System
AIX 4.3.3 + SL 8
IBM Edge Server V2 Load Balancer
IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime
Windows 2000 + SP2
IBM WebSphere Application Server Advanced Edition V4.0
DB2 UDB EE 7.2
IBM HTTP Server 1.3.19

Figure 5-7 Load balancer hot standby Product mapping option 1

The Edge Server Load Balancer components keep in constant contact with each other by exchanging a heartbeat between them. This enables the standby load balancer to take over without losing IP packets.

Product mapping option 2

The Product mapping shown in Figure 5-8 provides a load balancer hot standby configuration without increasing the need for physical machines.



Outside world
DMZ
Internal network
Windows 2000 + SP2
IBM Edge Server V2 Load Balancer
IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3) developer kit and runtime
Public Key Infrastructure
Domain Name Server
User
INTERNET
Protocol Firewall
Load Balancer
Or
Web Application Server
Backup LB
Web Application Server
Domain Firewall
Directory and Security Services
Database
Shared File System
AIX V4.3.3 + SL 8
IBM Edge Server V2 Load Balancer
IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime
AIX V4.3.3 SL 8
IBM WebSphere Application Server Advanced Edition V4.0
DB2 UDB EE 7.2
IBM HTTP Server 1.3.19
IBM Edge Server V2 Load Balancer
IBM AIX Developer kit, Java 2 Technology Edition, V1.3.0 for runtime

Figure 5-8 Load balancer hot standby Product mapping option 2

IBM Edge Server allows you to run a backup Load Balancer on one of the Web application server machines. This is referred to as *collocation*. Collocation of the WebSphere Edge Server Load Balancer component is supported on AIX®, Linux®, and Solaris™ platforms.

In this configuration, the backup load balancer node could be affected by problems on the Web Application Server node. In addition, collocation of the backup load balancer function on a content server machine is not supported on all platforms.

The following are a list of defined Runtime patterns for high availability:

- Basic Runtime pattern
- Runtime pattern variation 1: Single load balancer
- Runtime pattern variation 2: Load balancer hot standby
- Runtime pattern variation 3: Mutual high availability
- Runtime pattern variation 4: Wide area load balancing

Performance factors

There are two distinct aspects to performance. One is the minimum amount of time needed to process a given task. This is based on the nature of the task and the application design. The other is maintaining the amount of time needed for each task if the number of requests for the task increases.

Performance is the result of three factors that interact with each other:

- Time
- Resources
- Workload

Runtime patterns for performance

For the Runtime patterns for performance, we start off with one of the Runtime variations from the High Availability Runtime patterns, specifically Runtime pattern variation 2: load balancer hot standby. Refer to Figure 5-6 on page 150 for a graphical representation. Figure 5-9 shows a Runtime pattern for performance, Runtime pattern variation 5: Redirectors.

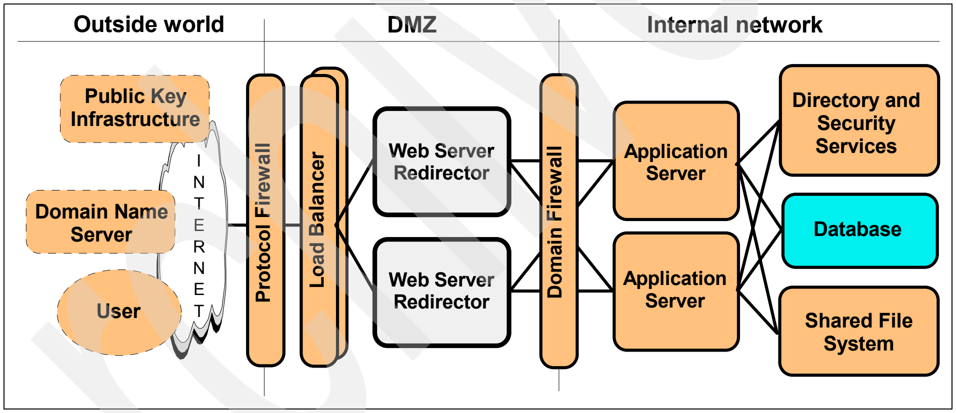


Figure 5-9 Runtime pattern variation 5: Redirectors

The Web server redirector node decides whether the request is being served by the local HTTP server or forwarded to the application server nodes. This Runtime pattern moves the application server nodes behind the domain firewall, adding further security.

The Web server redirector node is capable of performing workload management for the requests targeted at the application server nodes in the secure network. In addition, it can perform a port change to match the ports on which the application server nodes are listening. It also takes care of cookie-based server affinity.

This variation has the following benefits:

- Each node type (Redirector and Application Server) can be configured to the specific tasks.
- A domain firewall between the two node types adds additional security.
- More flexibility in the number of nodes of each type.
- Flexibility to not apply security on certain static content.

This variation has the following disadvantages:

- Many configuration changes on the Application Server node call for a redeployment of the plugin-cfg.xml file to all Web server redirector nodes.
- The configuration of the domain firewall requires additional skills.

The Product mapping for the Runtime pattern variation 5: Redirector is shown in Figure 5-10.

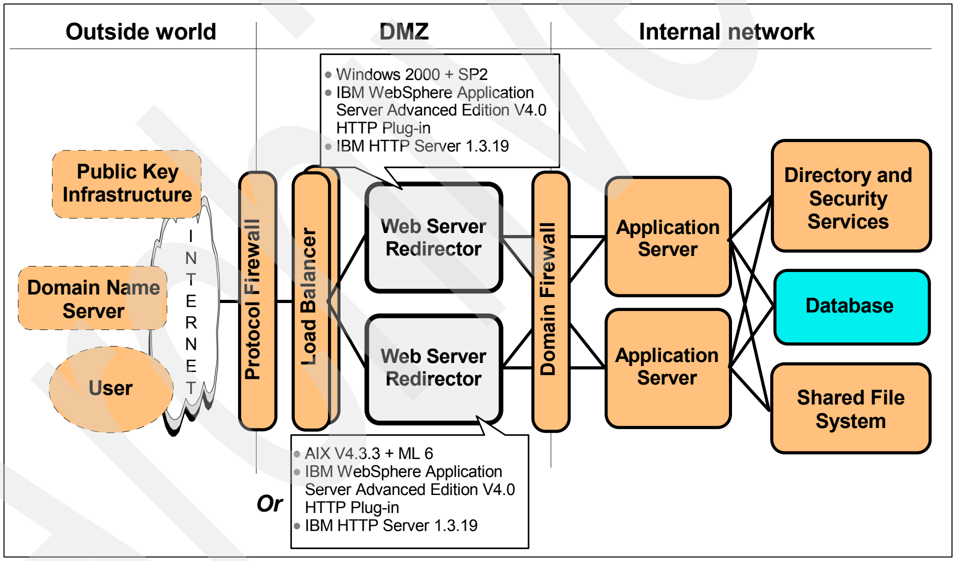


Figure 5-10 Product mapping, Redirector

The Web server redirector nodes run the IBM HTTP Server with the WebSphere Application Server HTTP plug-in. Again, notice that two redirector nodes are illustrated in Figure 5-10. One redirector node could possibly be sufficient to satisfy all requests. However, high availability considerations recommend using two redirectors with the load balancer spraying requests to both of them.

The following is a list of defined Runtime patterns for Performance:

- Runtime pattern variation 5: Redirectors
- Runtime pattern variation 6: Separation
- Runtime pattern variation 7: Caching proxy

For more detailed information, including technology and security guidelines, refer to the IBM Redbooks publication *Patterns for the Edge of Network*, SG24-6822.

5.2 Summary

These existing custom designs will help you determine the high-level shape of a WebSphere/Domino, WebSphere/SAP, or WebSphere/Edge Server integration solution. By starting with reasonably complete architectures, you save considerable development time and obtain assurance that the end solution will have a much higher chance of success. Actual savings will vary from project to project, but reduction of work effort in their design and architecture phases alone is of great benefit.

5.2.1 Key terms and concepts

"Glossary" on page 203 provides definitions of key terms and concepts.

5.2.2 Questions

Answer the following questions. To check your answers, see "Chapter 5: Answers to questions" on page 195.

5.2.3 Discussion and review questions

1. Describe how the Self-Service business pattern solution is commonly observed as:
   - An e-business solution
   - A collaboration business pattern solution

   Give an example of an application utilizing these two patterns.
2. List and elaborate on the characteristics that indicate a need for a WebSphere/Domino Integration design.

5.3 Exercise and challenge questions

Answer the following exercise and challenge questions.

5.3.1 Challenge questions

1. We use an older hybrid Runtime pattern for WebSphere and Domino. Review the newest Runtime patterns (see “Hybrid Runtime patterns” on page 141 for resources). How do these compare to the Runtime pattern utilized in the text?
2. Hershey Foods implemented an SAP rollout that had a significant impact on their organization, much further than simply IT. Research the project and provide an oversight into the issues that occurred during the rollout. What impact, specifically, was incurred on the production lines? How long did the rollout process take? Can you offer suggestions for how some problems could have been avoided?
3. How can an organization justify placing one server in hot standby mode from a feasibility standpoint? Refer to Figure 5-6 on page 150 for a Runtime pattern illustration. What type of industry or organization can justify such a solution?

5.3.2 Team case study projects

- “Walk Through and Learn Case Study 1: JPA Membership Services”, found in *Patterns: Custom Designs for Domino & WebSphere Integration*, SG24-6903)

  The JPA Membership Services case describes a simplified version of an actual application created for a Japanese professional association (JPA) that wished to publish its existing and future member information on the internet.

  The JPA has around 20,000 members. It currently manages its membership with a Lotus® Domino application and Lotus Notes® clients. Lotus Domino was used initially due to its workflow capabilities. The current environment is required because professionals in certain fields must be registered to practice in Japan. However, the information maintained is not currently used to support the individuals in achieving market exposure. The primary goals for implementing the new system are as follows:

  a. Make it easier for the public to find and contact members through the Internet.

  b. Add value to members by providing each member with their own home page with a virtual address that they can use on their own stationery and marketing material.

  c. Provide 24 x 7 access for the public to find members, rather than only during JPA office hours, as is currently the case.

After walking through and learning from the case study, as a team, discuss what you learned by identifying the emerging principles and patterns.

- “Walk Through and Learn Case Study 2: Laura's Gadgets e-shop”, found in *Patterns: Custom Designs for Domino & WebSphere Integration*, SG24-6903

  The Laura's Gadgets e-shop case describes a simplified version of an actual application created for a successful regional retailer that specializes in the sales of connected-ware, such as PDAs and mobile phones. The company wishes to provide its customers with an online sales site, integrate the site with existing back-end systems, and automate more of its supply chain function.

  Laura's Gadgets e-shop currently has several shops that sell mobile phones, PDAs, portable audio devices, and accessories. The current IT environment includes:

  a. A product catalog management system based on Domino

  b. A stock inventory and supply application

  c. Bank-supplied terminals for processing credit card settlements

  The new e-shop application is expected to provide:

  a. Quick Web browser-based product searches

  b. Customer registration

  c. Shopping cart and order tracking for customers

  d. Integration with the existing catalog and inventory systems

  e. Web browser-based credit card settlements

  After walking through and learning from the case study, as a team, discuss what you learned by identifying the emerging principles and patterns.

- “Walk Through and Learn Case Study 3: Business Loan Applications, found in *Applying the Patterns for e-business to Domino and WebSphere Scenarios*, SG24-6255.

  The Business Loan Applications case describes an Australian bank's business loan application process and a solution designed to streamline it.

  The existing loan process is a labor intensive one, which can require numerous interactions between the banking staff and the business or its representatives. Since the procedure can involve many parties, it is not easy for the customer to get a sense of what stage the application is at.

The objectives for the new loan application management process are as follows:

a. Simplify the data entry and allow the customer to self-administer it where possible.

b. Ensure the process is readily repeatable and easier to track.

c. Allow the customer to track the status of their application.

d. Expedite detailed financial queries by using collaboration facilities.

e. Utilize staff more effectively by scheduling the work and loan applicant appointments.

After walking through and learning from the case study, as a team, discuss what you learned by identifying the emerging principles and patterns.

- “Walk Through and Learn Case Study 4: HR Staffing Inc.”, found in *Patterns: Custom Designs for Domino & WebSphere Integration*, SG24-6903.

  The HR Staffing Inc. case describes a growing medium-sized job placement company that has had success in its traditional brick and mortar business. The company now wishes to connect its existing systems to the internet, replacing its current Web site, to provide higher service levels for the job seekers and job providers it services.

  HR Staffing's goals are as follows:

  a. Improve organizational efficiency by reducing the time required to process résumés and applications.

  b. Reduce costs by electronically processing this information.

  After walking through and learning from the case study, as a team, discuss what you learned by identifying the emerging principles and patterns.

Citizens Property Insurance Corp., Florida's largest home insurer, modernizes to handle drastic spikes in activity

Please refer to the following URL to learn more about this topic:

http://www-306.ibm.com/software/success/cssdb.nsf/CS/GJON-6TUSGR?OpenDocument&Site=default&cty=en_us

Case study

Here we discuss the case study of Citizens Property Insurance Corp.

Overview

Citizens Property, which provides all-perils insurance to Florida residents unable to find coverage in the private market, stepped up from an aging, error-prone processing system to a Web-based solution from IBM Software as Services, hosted by IBM Global Services - e-Business Hosting Services.

Business need

Citizens Property, state-sponsored insurer of last resort, was growing faster than anticipated as storm disasters increased and insurance companies fled the state or collapsed. The quasi-governmental company needed to replace its inefficient policy and claims administration system with a modern processing environment that reduced errors, increased underwriting efficiency, and afforded remote access to agents in the field.

Key challenges

Subject to the vagaries of storm activity and fluctuation in the availability of private-market coverage, Citizens needed a solution that could handle unpredictable patterns of demand at reasonable cost, and one that could provide a safe haven for its host system.

Solution: IBM Software as Services

Citizens Property chose an IBM Software as Services solution: customized software from BIPT, an IBM Business Partner, delivered remotely as needed by IBM e-business Hosting™ Services.

Web-based TiVA, BIPT's flagship insurance processing solution built on IBM WebSphere, DB2 and Lotus Domino, integrates property/casualty insurance rating and quotation, underwriting, claims, and accounting management functions. The IBM host server running TiVA is far removed from where most hurricanes strike.

Results

Filing applications and claims through the Web has cut processing costs and reduced manual errors, bringing the Citizens Property policy issuance cycle down from weeks to days. Underwriting is more accurate because the new solution permits field agents to interface remotely with insurance service bureaus that provide valuable land surveys and audits of particular locations.

Citizens Property pays only for the computing and hosting capacity it actually uses, helping the company manage unpredictable demand patterns and holding down costs for the insurer and its policy holders. And, as with other offerings in the IBM Software as Services portfolio, the client benefits from minimal upfront expense, rapid implementation, reduced risk, and lower costs aligned with usage.

Customer information

Citizens Property Insurance Corp. is a quasi-governmental company created by the Florida Legislature to serve as the state's insurer of last resort, offering all-perils homeowners insurance to any Florida resident who cannot find coverage in the private market.

- Case Study 5: Citizens Property Insurance Corp., Florida's largest home insurer, modernizes to handle drastic spikes in activity.

  Team assignment:

  Based on the information communicated in the Citizens Property case, collaboratively perform the following:

  a. Identify the Business and IT drivers.

  b. Identify the requirements.

  c. Documents the client environment.

  d. Define the solution by selecting the appropriate patterns.

  Perform a team project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

TransAlta powers up with IBM WebSphere Portal and SAP Enterprise Portal

Please see the following URL for information about this topic:

```
http://www.ibm.com/software/success/cssdb.nsf/cs/fwas-6b7c3x
```

The challenge

Provide integrated business information; Enhance enterprise communication and collaboration; Simplify enterprise-wide user experience; Reduce costs of doing business

The solution

Industry	Power generation
Application	SAP R/3 Enterprise™
Hardware	IBM eServer™ pSeries®
Software	IBM WebSphere Portal, IBM WebSphere Business IntegrationServer, IBM WebSphere MQ Workflow, SAP R/3, SAP Business, Information Warehouse, and SAP Enterprise Portal
Services	IBM Global Services and IBM Software Services

The benefits

Provides a single access point for users of SAP applications, Lotus Domino, and desktop productivity applications, single user sign-on for all services within TransAlta, and reduced IT department maintenance and system costs with fewer interfaces to maintain

Founded in 1911, TransAlta is the largest non-regulated power generation and wholesale marketing company in Canada. TransAlta has annual revenues of more than C$2.5 billion, with operations in Australia, Canada, Mexico, and the US. TransAlta has used SAP since 1995, replacing some 40 disparate applications with a single, unified view of enterprise-wide information and business processes. In 2000, TransAlta migrated its SAP systems to IBM eServer pSeries servers, configured in a High Availability Clustered Multi Processing environment. In short, SAP and IBM form the foundation of TransAlta's business information strategy.

To meet this objective, TransAlta created a single business portal that allows users access to SAP, Lotus Domino and, as a bonus, standard desktop office applications, all from the same single-user environment and sign-on point. The challenge was to provide portal access to such a diverse range of data and applications, and to achieve this TransAlta has exploited both and IBM WebSphere Portal and SAP Enterprise Portal in a dual-portal strategy, supported by multiple IBM WebSphere integration technologies.

The support of the IBM SAP Collaboration Technology Support Center (CTSC) during the architectural design phase was a very important asset in the portal project.

Integrated portals reduce costs

Portal technology means that users need only have browser access to be able to view, manage, and enter data into SAP, Domino, and many other applications. The consistent interface for multiple programs means training costs are low, and ease of use and productivity is high. Integrating SAP R/3 data, Lotus Domino collaboration and teamwork applications, and office productivity programs data proved to be a tough integration task.

The WebSphere Portal integrates the Lotus Domino and collaboration applications into a single user experience. The look-and-feel of the WebSphere Portal has been tailored to the precise look-and-feel of the SAP Enterprise Portal, so closely, in fact, that the dual portal appears to be a single interface to users, thanks to the flexibility of the IBM WebSphere Portal. From within the TransAlta dual portal, users are able to reach all their relevant SAP applications, as well as log in to Notes e-mail, calendars, QuickPlace, SameTime, and other Domino applications. A single user login spanning over the WebSphere Portal, the SAP Enterprise Portal, and all SAP applications grants access to relevant application areas without the need to re-authenticate from system to system. In addition, the user gets a consistent interface regardless of the application being used.

Two complex landscapes have been combined in one integrated, simplified platform. The integration framework combines both technologies on presentation layer, which allows seamless integration for the user. Before WebSphere Portal was introduced, users would have to jump between applications to find data, or send faxes or make calls. By being able to move between underlying applications while inside the portal, users are able to find and act on the information they need in a more timely fashion. This cuts the costs of doing business by reducing stores of paper-based information, such as events, news, staff directory, self-serve human resources materials, procedure manuals, and reference documents.

Support from IBM Business Partners

The TransAlta Enterprise Portal project was developed using IBM Global Services, IBM Software Services, the CTSC, and IBM Business Partner Sandbox Systems. Sandbox produced a dual portal proof-of-concept in three weeks that convinced TransAlta of the viability of the approach of a dual Portal approach. The speed of production and low maintenance cost of the WebSphere Portal, reducing support staff assigned to the project from about 20 to around five, convinced the TransAlta team that WebSphere offered the right solution.

Workflow

TransAlta's next ambition is to combine portal access to information with workflow capabilities, allowing users to work their business processes through the portals. This entails being able to tap into operational systems, and provide information directly to the portal. To achieve this, TransAlta has selected IBM WebSphere MQ Workflow, and IBM as its integration partner.

- Case Study 6: TransAlta powers up with IBM WebSphere Portal and SAP Enterprise Portal. (Your team may use, for reference purposes, the following IBM Redbooks publication: *WebSphere Business Integration for SAP*, SG24-6354).

  Team assignment:

  Based on the information communicated in the TransAlta case, collaboratively perform the following:

  a. Identify the Business and IT drivers.

  b. Identify the requirements.

  c. Document the client environment.

  d. Define the solution by selecting the appropriate patterns.

Perform a team project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

Nextra banks on IBM portal solution to win and retain customers

Please refer to the following URL for information about this case:

http://www.redbooks.ibm.com/additional_materials/SG247466/Nextra_-_IBM_Portal_Solution.pdf

- Case Study 7: Nextra banks on IBM portal solution to win and retain customers.

  Team assignment:

  Based on the information communicated in the Nextra case, collaboratively perform the following:

  a. Identify the Business and IT drivers.

  b. Identify the requirements.

  c. Document the client environment.

  d. Define the solution by selecting the appropriate patterns.

  Perform a team project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

Kookmin Bank heightens customer responsiveness with IBM software

Please refer to the following URL for information about this case:

http://www.redbooks.ibm.com/additional_materials/SG247466/Kookmin_Bank.pdf

- Case Study 8: Kookmin Bank heightens customer responsiveness with IBM software

  Team assignment:

  Based on the information communicated in the Kookmin Bank case, collaboratively perform the following:

  a. Identify the Business and IT drivers.

  b. Identify the requirements.

  c. Document the client environment.

  d. Define the solution by selecting the appropriate patterns.

Perform a team project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

Volkswagen's world-class procurement strategy produces breakthrough productivity gains

Please refer to the following URL for information about this case:

http://www.redbooks.ibm.com/additional_materials/SG247466/Volkswagen_World-class_procurement_strategy.pdf

- Case Study 9: Volkswagen's world-class procurement strategy produces breakthrough productivity gains.

  Team assignment:

  Read the Real-World Environment Case found in the beginning of Chapter 5, "Volkswagen's world-class procurement strategy produces breakthrough productivity gains." Collaboratively perform the following:

  a. Identify the Business and IT drivers.

  b. Identify the requirements.

  c. Document the client environment.

  d. Define the solution by selecting the appropriate patterns.

  Perform a team project debriefing to gain insights and a better understanding of the steps and the application of the patterns approach.

5.3.3 Multiple choice

1. Patterns for e-business has identified _____ as an existing custom design that can be exploited as a reusable asset.

   a. WebSphere and Domino Integration.

   b. WebSphere and SAP Integration.

   c. WebSphere and Edge Server Integration.

   d. All of the above.

2. _____ combine the transactional functionality of the Self-Service business pattern with the collaborative functionality of the Collaboration business pattern.

   a. WebSphere and Domino Integration.

   b. WebSphere and SAP Integration.

   c. WebSphere and Edge Server Integration.

   d. All of the above.

3. The _____ business pattern is commonly observed in e-business solutions that provide users the ability to access their information and change it by interacting directly with core business systems and databases.

   a. Self-Service.

   b. Collaboration.

   c. Both A and B are correct.

   d. Neither A or B is correct.

4. The _____ business pattern occurs in e-business solutions that involve one- or two-way interactions between users of the solution.

   a. Self-Service.

   b. Collaboration.

   c. Both A and B are correct.

   d. Neither A or B is correct.

5. Hybrid Runtime patterns are referred to as such because they:

   a. Show both middleware nodes commonly seen on a normal Runtime pattern.

   b. Show a graphical representation of the Business and Integration patterns used in combination with one another to create this solution design.

   c. Both A and B are correct.

   d. Neither A or B is correct.

6. The _____ design represents multiple different combinations of the following patterns: Self-Service, Information Aggregation, and Application Integration.

   a. SAP/Integration Custom.

   b. Electronic commerce for large distributor/retailer enterprise.

   c. Both A and B are correct.

   d. Neither A or B is correct.

7. Runtime patterns:

   a. Describe the logical architecture required to implement an Application pattern.

   b. Depict the major middleware nodes, their roles, and the interfaces between the nodes.

   c. Address the processing logic and data placed on the middleware nodes.

   d. All of the above.

8. The rule for High Availability for Runtime environments is:

   a. Every entity must be redundant.

   b. Each entity must be monitored for failure.

   c. Both A and B are correct.

   d. Neither A or B is correct.

9. The rule for Performance for Runtime environments is:

   a. Every entity must be redundant.

   b. Each entity must be monitored for failure.

   c. Both A and B are correct.

   d. Neither A or B is correct.

10._____ is a distinct aspect of performance.

   a. The maximum amount of time needed to process a given task.

   b. Maintaining the amount of time needed for each task if the number of requests for the task increases.

   c. Both A and B are correct.

   d. Neither A or B is correct.

5.3.4 Fill in the blank

1. The _______________________ Business pattern is commonly observed in e-business solutions that provide users the ability to access their information and change it by interacting directly with core business systems and databases.

2. The _________________ Integration Custom design represents multiple different combinations of the Self-Service and Collaboration Business patterns, along with the Access and Application Integration patterns.

3. Many companies need to address the integration of high availability and _____________ into their e-business solutions.

4. Runtime patterns demonstrate Service level characteristics, specifically _______________ requirements such as availability, performance, scalability, and security.

5. ____________________ patterns are divided into two categories: high availability and performance.

6. The ________________ pattern removes the single point of failure from the load balancer node by adding a second node with the same function.

5.3.5 True or false

1. Because most enterprises have similar e-business needs, successful solutions can often be reused.
2. Business and IT drivers are a consistent set of objectives that assist the architect in selecting the appropriate Business and Integration patterns.
3. Navigation of the hybrid Runtime patterns performed as a linear review is the quickest method to navigate through the patterns.
4. Navigation of the hybrid Runtime patterns performed through the matrix for navigation through the patterns requires an understanding of the required functionality, complexity, and extensibility of the solution being designed.
5. The Patterns for e-business designs cannot be implemented using any software component other than IBM products.
6. Runtime patterns are divided into two categories: high availability and performance.
7. Runtime patterns are shown in graphical form using standard node types to identify the function represented by that node.
8. One limitation of utilizing the Runtime pattern variation with load balancer on hot standby is that one server is always in hot mode, doing no productive work.

5.3.6 Short answer

1. Describe how the Self-Service business pattern solution is commonly observed as
   - An e-business solution
   - A collaboration business pattern solution

   Give an example of an application utilizing each of these two patterns.
2. List and elaborate on the characteristics that indicate a need for a WebSphere/Domino Integration design.
3. Why would an organization plan an SAP rollout for all subsidiaries to be implemented over a five year period?
4. List the benefits of implementing the Runtime pattern that utilizes load balancer hot standby.
5. What are the advantages and disadvantages of implementing the Runtime pattern variation 5: Redirectors?

A

Answers

This appendix provides the answers to the questions listed at the end of each chapter throughout this IBM Redbooks publication. The answers are broken down by chapter.

Chapter 1: Answers to questions

Here we give the answers to the questions asked in Chapter 1, "Introduction to Patterns for e-business and various pattern approaches" on page 1.

Answers to the discussion and review questions in 1.6, "Questions" on page 18

1. A *pattern* can be defined as a model or plan that used as a guide in making things. As such, patterns serve to facilitate the development and production of things. They codify the repeatable experience and knowledge of people who have performed similar tasks before. They not only document solutions to common problems, but also point out pitfalls that should be avoided.
2. Each of the patterns helps companies understand the true scope of their development project and provide the necessary tools to facilitate the application development process, thereby allowing companies to shorten time to market, reduce risk, and more importantly, realize a more significant return on investment.
3. The Patterns for e-business approach is based on a set of layered assets that can be exploited by any existing development method. These layered assets are structured in such a way that each level of detail builds on the last. These assets include:

    a. Business patterns that identify the interaction between users, businesses, and data.

    b. Integration patterns that tie multiple Business patterns together when a solution cannot be provided based on a single Business pattern.

    c. Composite patterns that represent commonly occurring combinations of Business patterns and Integration patterns.

    d. Application patterns that provide a conceptual layout describing how the application components and data within a Business pattern or Integration pattern interact.

    e. Runtime patterns that define the logical middleware structure supporting an Application pattern. Runtime patterns depict the major middleware nodes, their roles, and the interfaces between these nodes.

    f. Product mappings that identify proven and tested software implementations for each Runtime pattern.

    g. Best-practices guidelines for design, development, deployment, and management of e-business applications.

4. A Custom design may also be a Composite pattern if it recurs many times across domains with similar business problems.

5. Some ways are described in the text (provided below). Each of these initiations may include a high level use case.

   a. An executive vision statement that indicates a need for business partners to have a single company interface. This is an implicit invitation to spend time understanding the customer organization and uncovering opportunities for software and hardware products to create this vision.

   b. A response to an RFP from customers through the formal proposal process, which is frequently utilized in the public sector.

   c. A response to "Line of Business" executive (CRM, SAP, or WebSphere) requests, such as the ability to improve responsiveness to an account inquiry by making simple inquiries available to the customer and provide a starting point for future e-business development. This request is an invitation to start investigating a core part of the operations and use the Custom design icon view to decompose this business issue.

   d. A requested technical solution (such as a middleware requirement) to integrate systems (for example, WebSphere and Domino systems). Requests such as these are common and the use of e-business Patterns may shorten the steps in the integration methodology.

Answers to exercise and challenge questions in 1.7, “Exercise and challenge questions” on page 18

1. Select a Business/Integration/Composite pattern or a Custom design.

   Business patterns for Simple Implementations

   Business patterns are high-level constructs that can be used to describe the key business purpose of a solution. These patterns describe the objectives of the solution, the high-level participants that interact in the solution, and the nature of the interactions between the participants. Structurally, these patterns are made up of at least two of the following three entities that occur in e-business solutions:

   – Users of the solution: This can include customers, investors, partners, vendors, and so on.

   – Enterprise or organization the users interact with: This "Business" entity can be used to represent the organization itself or systems (applications or software programs) that exist within the organization.

   – Data that exists within the organization: Data is distinguished from applications because the nature of interactions between these entities is very different.

These three Business patterns highlight the most commonly observed interactions between Users, Businesses, and Data. They are the fundamental building blocks of most e-business solutions, and have the following characteristics:

- Each pattern is self-contained. The scope of each pattern embraces the minimum end-to-end flows necessary to implement an automated business process. For example, the Self-Service business pattern designs solutions that include an end-to-end transaction flow, a security flow, and a restart-recovery flow.
- Each pattern typically interacts with other patterns through one or more integration points. These integration points might include file transfer, message transfer, a common data base, common components, common applications, common processes, common access points, or a common workflow.

Self-Service: Also known as the User-to-Business pattern, Self-Service addresses the general case of internal and external users interacting with enterprise transactions and data.

Collaboration: Sometimes called User-to-User, the Collaboration Business pattern addresses the interactions and collaborations between users. This pattern can be observed in solutions that support small or extended teams who need to work together in order to achieve a joint goal.

Information Aggregation: The Information Aggregation Business pattern, also known as User-to-Data, can be observed in e-business solutions that allow users to access and manipulate data that is aggregated from multiple sources. This Business pattern captures the process of taking large volumes of data, text, images, video, and so on and using tools to extract useful information from them.

Extended Enterprise: The Extended Enterprise Business pattern (also known as Business-to-Business or B2B) addresses the interactions and collaborations between business processes in separate enterprises. This pattern can be observed in solutions that implement programmatic interfaces to connect inter-enterprise applications.

2. The Composite patterns

   When Business patterns and Integration patterns are combined to assemble solutions that perform complex business functions, certain recurring pattern combinations begin to emerge. For example, most solutions today provide a personalized user experience. These solutions typically expose key functions of line-of-business applications by integrating with them. Consequently, most of these solutions combine an Access Integration pattern, a Self-Service pattern, and an Application Integration pattern to enable this complex functionality. Composite patterns represent commonly occurring combinations of Business patterns and Integration patterns such as this one, and typically solve major portions of functionality within a solution.

The Patterns for e-business are designed for reuse in nearly any Web-based application development scenario. As such, the possible combinations of Business patterns and Integration patterns that form Composite patterns are extensive. The following, however, lists some of the most commonly occurring Composite patterns.

- Electronic Commerce: Electronic Commerce combines the Self-Service and Information Aggregation Business patterns with the Application Integration pattern to facilitate user interaction. Additionally, an electronic commerce site can integrate supply chain management functionality into the solution through the Extended Enterprise pattern, or e-mail confirmation of orders to customers through the Collaboration pattern.
- e-Marketplace: e-Marketplaces are trading exchanges, sell-side hubs, and buy-side hubs that facilitate and promote buying, selling, and business communities among trading partners within certain industries. These solutions represent some of the most comprehensive and complex e-business applications that exist today.
- Portal: Though designed to facilitate many variations of similar functionality, a Portal solution typically aggregates multiple information sources and applications to provide a single, seamless, and personalized access to users.
- Account Access: Account access solutions provide customers around-the-clock access to their account information. They also allow users to inquire, update, and delete information on their individual accounts. Many applications fall under this category of solutions, ranging from trading applications provided by online brokerages to account manager functions provided by utilities such as telephone companies.

The difference is that several of the patterns identified in the text are not given on the Web site, also on the Web site; e-Marketplace is new to the list. If the student reads further, he or she will note that the e-Marketplace Composite pattern has the same definition as the Sell-Side Hub (Supplier) Composite pattern identified in the text. Students should be able to identify the differences. Also note that this answer will change as the Web site is altered.

Team Project writing project and presentation in 1.7.2, “Team project” on page 19

- Teams of three or four writing an assignment and presentation:
  - Rationale for the Patterns for e-business
  - Characterization of Patterns

Answers to multiple choice in 1.7.3, “Multiple choice” on page 20

1. D
2. A
3. A
4. B
5. C
6. B
7. C
8. A
9. B
10. A
11. C
12. D
13. A
14. B
15. B
16. B
17. C
18. A
19. A
20. D
21. C

Answers to fill in the blank in 1.7.4, “Fill in the blank” on page 24

1. IBM Patterns for e-business
2. Patterns or Patterns for e-business
3. layered assets
4. Integration patterns
5. Product mappings
6. User-to-Business
7. User-to-Data

8. Custom design
9. Sell-Side Hub (Supplier)

Answers to true or false in 1.7.5, “True or false” on page 25

1. True
2. True
3. False
4. False
5. True
6. False
7. False
8. True

Answers to short answer in 1.7.6, “Short answer” on page 25

1. A *pattern* can be defined as a model or plan used to guide in making things. As such, patterns serve to facilitate the development and productions of things. They codify the repeatable experience and knowledge of people who have performed similar tasks before. They not only document solutions to common problems, but also point out pitfalls that should be avoided.
2. Some ways are described in the text (provided below). Each of these initiations may include a high level use case:
   a. An executive vision statement that indicates a need for Business Partners to have a single company interface. This is an implicit invitation to spend time understanding the customer organization and uncovering opportunities for software and hardware products to realize this vision.
   b. A response to an RFP from customers through the formal proposal process that is frequently utilized in the public sector.
   c. A response to "Line of Business" executive (CRM, SAP, or WebSphere) requests, such as the ability to improve responsiveness to an account inquiry by making simple inquiries available to the customer and provide a starting point for future e-business development. This request is an invitation to start investigating a core part of the operations and use the Custom design icon view to decompose this business issue.

A requested technical solution such as a middleware requirement to integrate systems (for example, WebSphere and Domino systems). Requests such as these are common and the use of e-business Patterns may shorten the steps in the integration methodology.

Chapter 2: Answers to questions

Here we give the answers to the questions asked in Chapter 2, “Business Solution Delivery methods” on page 27.

Answers to the discussion and review questions in 2.4, “Questions” on page 38

1. What is a method family? How do engagement models relate to method families?

   The IBM Global Services Method is a cornerstone of IBM Global Services' goal of shifting its business model. It is organized around a number of method families that reflect the close relationships between a number of engagement models applicable to the same domain, such as e-business, application maintenance, and so on. Engagement methods describe typical approaches to delivering a project in terms of a work breakdown structure to be adhered to, the work products to be created, the techniques to be applied, and the roles to be allocated to the project.

2. What is the relationship between work products and deliverables?

   Work products are not the same thing as deliverables. Work products are tangible. The term deliverable is used to refer to a document which contractually needs to be delivered to the customer, which is not necessarily the case for work products.

3. Provide a brief description of the events that occur in the phases of the e-business Custom Engagement Model process provided in Figure 2-2 on page 35:

   a. Solution Startup: Covers the required startup activities. Focuses on the activities and tasks needed to define, plan, and start an engagement. The focus of the activities is to tailor the engagement model in order to determine the actual phase/activity/task structure that will be followed during the project and work products that will be created throughout the project.

   b. Solution Outline: Provides the client with the necessary costs, schedule, and risk information to make an informed investment decision regarding a potential new system. The high level business requirements, business objectives, and IT objectives are taken into consideration, focusing each time on those requirements/objectives influencing the overall architecture.

c. Macro Design: Develops a robust architectural framework upon which to build agile releases. The overall architecture of the IT system is specified, which allows definition of solution-wide approaches for addressing both functional and non-functional requirements.

d. Micro Design: Prepares for the build cycle of a specific release of the system by driving the architecture design to a release- and implementation-specific view.

e. Build Cycle: Incrementally develops and tests the system until the objectives of the release are achieved. Within a specific release, multiple Build Cycles could be planned, each with its own objectives in line with and contributing to the overall Release specific objectives.

f. Deployment: Deploys the system and prepares for the next release of the system.

g. Solution Close: Covers the required close activities. The results of the project are assembled, analyzed, and archived for use by other practitioners

4. What is a Work Product Description (WPD)? How is a WPD organized?

   WPD describes a particular type of work product and gives guidance on how to create the actual work, notation, and recommended structure of the work product itself. WPDs cover a broad range of work performed on engagements and have been further organized into a number of domains and sub domains to further categorize content where needed.

Answers to exercise and challenge questions in 2.5, “Exercise and challenge questions” on page 39

Here we give the answers to the questions asked in 2.5, “Exercise and challenge questions” on page 39

Answer to challenge question in 2.5.1, “Challenge question” on page 39

Research IBM Global Services' goal of shifting its business model. How does the IBM Global Services Method tie into this organizational goal?

Answers to this question will vary, based on the research conducted by the student. In the text, the IBM Global Services' goal is stated as a shifting of the business model towards the delivery of asset-based services by providing the mechanism for practitioners to systematically reuse knowledge and assets using a consistent and integrated approach. The IBM Global Services Method is a cornerstone of this organizational goal. It is organized around a number of method families that reflect the close relationships between a number of engagement models applicable to the same domain, such as e-business, application maintenance, and so on.

Answers to Walk Through and Learn Case Study in 2.5.2, "Team project: Walkthrough and Learn Case Study" on page 40

- Ask each team to capture the insights gained as the team walked through the case study.
- Each team presents their insights to the class.
- With the instructor facilitating the class discussion, sharing team insights with the class for further discussion provides yet another context for deeper understanding.

Answers to multiple choice in 2.5.3, "Multiple choice" on page 41

1. D
2. D
3. A
4. B
5. C
6. A
7. D
8. A
9. B
10. C
11. A
12. B
13. C
14. B
15. A
16. C
17. A
18. B
19. D

20. A
21. D
22. B
23. C

Answers to fill in the blank in 2.5.4, “Fill in the blank” on page 45

1. engagement models
2. method tailoring
3. domains
4. Organization
5. Engagement
6. smaller
7. Releases
8. Deployment
9. Solution Startup, Solution Close

Answers to true or false in 2.5.5, “True or false” on page 46

1. True
2. False
3. False
4. True
5. True
6. True
7. True
8. True
9. True
10. False

Answers to short answer in 2.5.6, “Short answer” on page 47

1. The IBM Global Services Method is a cornerstone of the IBM Global Services goal. It is organized around a number of method families that reflect the close relationships between a number of engagement models applicable to the same domain, such as e-business, application maintenance, and so on. Engagement methods describe typical approaches in delivering a project in terms of a work breakdown structure to be adhered to, the work products to be created, the techniques to be applied, and the roles to be allocated to the project.
2. A WPD describes a particular type of work product and gives guidance on how to create the actual work, notation, and the recommended structure of the work product itself. WPDs cover a broad range of work performed on engagements. They have been further organized into a number of domains and sub domains to further categorize content where needed.
3. A brief description of the events that occur in the phases of the e-business Custom Engagement Model process are:
   a. Solution Startup: Covers the required startup activities. Focuses on the activities and tasks needed to define, plan, and start an engagement. The focus of the activities is to tailor the engagement model to determine the actual phase/activity/task structure that will be followed during the project and work products that will be created throughout the project.
   b. Solution Outline: Provides the client with the necessary cost, schedule, and risk information to make an informed investment decision regarding a potential new system. The high level business requirements, business objectives, and IT objectives are taken into consideration, focusing each time on those requirements/objectives influencing the overall architecture.
   c. Macro Design: Develops a robust architectural framework upon which to build agile releases. The overall architecture of the IT system is specified, which allows definition of solution-wide approaches for addressing both functional and non-functional requirements.
   d. Micro Design: Prepares for the build cycle of a specific release of the system by driving the architecture design to a release- and implementation-specific view.
   e. Build Cycle: Incrementally develops and tests the system until the objectives of the overall release are achieved. Within a specific release, multiple Build Cycles could be planned, each with their own objectives in line with and contributing to the overall Release specific objectives.
   f. Deployment: Deploys the system and prepares for the next release of the system.

g. Solution Close: Covers the required close activities. The results of the project are assembled, analyzed, and archived for use by other practitioners.

Chapter 3: Answers to questions

Here we give the answers to the questions asked in Chapter 3, “Building solutions on a Single Business pattern basis” on page 49.

Answers to discussion and review questions in 3.5.1, “Discussion and review questions” on page 71

1. Solution Startup Phase: Defining the project approach.

   Solution Outline Phase: Identifying requirements, concentrating on those that will have a significant impact on the solution. Defining the high level architecture to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be given to the more challenging areas of the proposed solution.

   Macro Design Phase: Identifying requirements to confirm the results of the Solution Outline Phase, with further refining of said requirements. Defining the architecture in terms of how it addresses the functional and non-functional requirements so that in subsequent phases, a start is made on its implementation. Building the development environment that includes the review of the product installation instructions in the applicable pattern to assist with the product installs for the development environment.

   Micro Design Phase: Refining the architecture to complete the design so that the build activities can be performed.

   Build Cycle Phase: Building the production environment based on the architecture defined in previous phases.

   Deploy Phase: Reviewing the solution plans and system management guidelines.

2. Question 2.

   a. Question A

      - The scope of the project is to Web-enable the existing CICS credit card application for account access, rewards programs, credit line increases, and so on.

b. Question B

- Reduction of time to market.
- Tactical focus on only supporting the Web channel.
- Strategic focus on supporting multiple access channels such as call center, telephone access, and so on.
- Leverage existing investment and the existing CICS skill base.

c. Question C

- A requirement to access an existing CICS-based application.
- A strong emphasis on the privacy of the exchanges between the front end and the existing applications and the authentication of a client before any access to an existing application is allowed.
- A preference for Windows 2000/IBM System x servers to host the Web infrastructure.
- A current installed base of either Windows 2000/System x or IBM System z servers.

3. Question 3.

a. Question A

The Self-service Business pattern is the Business pattern identified. This came about due to the interest in extending an existing credit card existing application through a Web channel to its existing customer/user base. This pattern is applicable if the requirements suggest that users interact directly with existing data.

b. Question B

The Router Application pattern is the Application pattern identified. This came about due to the intent to provide access to its existing back-end systems through different channels (initially only a Web channel, but the call center and telephone have been identified as future extensions). The Router Application pattern provides a structure for applications that require the intelligent routing of requests from multiple delivery channels to one of the multiple back-end applications.

c. Question C

The Router Runtime pattern is the Runtime pattern identified. This came about because the functions of the router tier are performed by an integration server.

4. Question 4.

   a. Question A

      - The overall objective is to gather the overall business requirements and objectives, particularly concentrating on those which will have a significant impact on the solution. This provides a more detailed description of requirements so that you can identify the factors and key business functions delivered by the proposed solution.

   b. Question B

      - The application must support multiple languages.
      - Performance must be enhanced over existing Web-based applications.
      - Capacity/logon (500 logons and fifty transactions every minute).
      - Scalability (10,000 users in two years).
      - Security (secure transaction support, strong encryption, and connections to CICS are allowed only if authenticated).
      - Technology constraints (front-end access is Netscape and IE browser-based, CICS V4.1 for others).

5. The Self-Service::Router Application pattern is the best fit. This Application pattern came from the review of the IT and Business drivers of the Self-service Business pattern. The Multiple Channel requirement maps clearly back to the key Business driver expressed by First Financial, and leveraging existing systems is one of their main IT related concerns. These two support the use of the Router Application pattern.

6. The router tier can route a single message to one or more applications within the same business process. This allows the router tier to process a single request within this business line and access the required systems to produce a response from corresponding systems, such as the rating and underwriting systems.

7. Question 7.

   a. Question A

   This activity concentrates on refining the system specifications based on all user requirements, business processes, and business requirements by elaborating the system requirements and specifications. It enables the team to make better estimates for the first release and provide an overall specification for the structure and topology of the system.

b. Question B

The overall objective of this activity during the Solution Outline phase is to gather the overall business requirements and objectives, particularly concentrating on those which will have a significant impact on the solution. This provides a more detailed description of requirements so that you can identify the factors and key business functions delivered by the proposed solution.

"Identifying requirements" during the Solution Outline phase referred to the activity of determining the key requirements influencing the overall architecture. "Identifying requirements" during the Macro Design phase concentrates on refining the system specifications based on all user requirements, business processes, and business requirements.

Answers to exercise and challenge questions in 3.6, “Exercise and challenge questions” on page 72

Here we give the answers to the questions asked in 3.6, “Exercise and challenge questions” on page 72.

Answers to challenge questions in 3.6.1, “Challenge questions” on page 73

1. Challenge question answers:
   - Estimating the man days needed to build new functionality or to integrate existing functionality.
   - Estimating the software budget, taking into account, among other elements, the suggested software products.
   - Specifying the resource profiles required to deliver the solution (among others derived from selected products).
   - Specifying the tooling required for development (among others derived from selected patterns).
2. The answer to this challenge will vary based on the student research conducted. When reviewing the findings, make sure the student can defend his or her findings based on research. Consider placing students in groups for research purposes, thus allowing you to assign one or more of the bulleted items above to specific groups.

Note: Bullets three and four above will require you to provide a foundation that identifies existing software for the First Financial organization, as these are not provided in the text.

Answers to team case study projects in 3.6.2, “Team case study projects” on page 73

For each case:

- Ask each team to capture the insights gained as the team collaboratively analyzed the case.
- Each team presents their solution and insights to the class.
- Discuss emerging themes and principles.
- With the instructor facilitating the class discussion, sharing team insights with the class for further discussion and reflection provides yet another context for deeper understanding.

Answers to multiple choice in 3.6.3, “Multiple choice” on page 76

1. D
2. A
3. B
4. C
5. C
6. C
7. B
8. B
9. B
10. A
11. B
12. D
13. A

Answers to fill in the blank in 3.6.4, “Fill in the blank” on page 79

1. User
2. Assets
3. Building the development environment
4. Micro Design
5. Refining the architecture

Answers to true or false in 3.6.5, “True or false” on page 80

1. True
2. False
3. False
4. True
5. True

Answers to short answer in 3.6.6, “Short answer” on page 80

1. Question 1.
   - Solution Startup Phase
   - Solution Outline Phase
   - Macro Design Phase
   - Micro Design Phase
   - Build Cycle Phase
   - Deploy Phase
2. Question 2.
   - Solution Startup Phase: Defining the project approach.
   - Solution Outline Phase: Identifying requirements, concentrating on those that will have a significant impact on the solution. Defining the high level architecture to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be given to the more challenging areas of the proposed solution.
   - Macro Design Phase: Identifying requirements to confirm the results of the Solution Outline Phase, with further refining of said requirements. Defining the architecture in terms of how it addresses the functional and non-functional requirements so that in subsequent phases, a start is made on its implementation. Building the development environment, which includes the review of the product installation instructions in the applicable pattern to assist with the product installs for the development environment.
   - Micro Design Phase: Refining the architecture to complete the design so that the build activities can be performed.
   - Build Cycle Phase: Building the production environment based on the architecture defined in previous phases.
   - Deploy Phase: Reviewing the solution plans and system management guidelines.

3. Question 3.

   a. The scope of the project is to Web-enable the existing CICS credit card application for account access, rewards programs, credit line increases, and so on.

      - Reduction of time to market.
      - Tactical focus on only supporting the Web channel.
      - Strategic focus on supporting multiple access channels, such as call center, telephone access, and so on.
      - Leverage investments and the existing CICS skill base.

   The second and third part of this question is opinion based. Look for the student's ability to defend choice. The preferred answer is the Strategic Focus to support multiple channels, based on the Router Application pattern that allows the system to route a single message to one or more applications.

      - A requirement to access an existing CICS-based application.
      - A strong emphasis on the privacy of the exchanges between the front end and the existing applications and the authentication of a client before any access to an existing application is allowed.
      - A preference of Windows 2000/System x servers to host the Web infrastructure.
      - A current installed base of either Windows 2000/System x or System z servers.

   The second and third parts of this question are opinion based. Look for the students to defend their choices. While the text does not explicitly provide a clear answer, we would suggest a preference for the first bullet, access to an existing CICS-based application based on the IT drivers to leverage existing systems, investment, and back-end application integration.

4. Question 4.

   The Self-service Business pattern is the Business Pattern identified. This came about due to the interest in extending an existing credit card application through a Web channel to its existing customer/user base. This pattern is applicable if the requirements suggest that users interact directly with data.

   The Router Application pattern is the Application Pattern identified. This came about due to the intent to provide access to its existing back-end systems through different channels (initially only a Web channel, but the call center and telephone have been identified as future extensions). The Router Application pattern provides a structure for applications that requires the intelligent routing of requests from multiple delivery channels to one of the multiple back-end applications.

The Router Runtime pattern is the Runtime Pattern identified. This came about because the functions of the router tier are performed by an integration server.

5. The overall objective is to gather the overall business requirements and objectives, particularly concentrating on those that will have a significant impact on the solution. This provides a more detailed description of requirements so that you can identify the factors and key business functions delivered by the proposed solution.

   – The application must support multiple languages.

   – Performance must be enhanced over existing Web-based applications.

   – Capacity/logon (500 logons and fifty transactions every minute).

   – Scalability (10,000 users in two years).

   – Security (secure transaction support, strong encryption, and connections to CICS are allowed only if authenticated).

   – Technology constraints (front-end access is Netscape and IE browser-based, CICS V4.1 for others).

6. The Self-Service::Router Application pattern is the best fit. This application pattern came about from the review of the IT and Business drivers of the Self-service Business pattern. The Multiple Channel requirement maps clearly back to the key Business driver expressed by First Financial, and leveraging existing systems is one of their main IT related concerns. These two support the use of the Router Application pattern.

7. This activity concentrates on refining the system specifications based on all user requirements, business processes, and business requirements by elaborating the system requirements and specifications. It enables the team to make better estimates for the first release and provide an overall specification for the structure and topology of the system.

   The overall objective of this activity during the Solution Outline phase is to gather the overall business requirements and objectives, particularly concentrating on those that will have a significant impact on the solution. This provides a more detailed description of requirements so that you can identify the factors and key business functions delivered by the proposed solution

   "Identifying requirements" during the Solution Outline phase referred to the activity of determining the key requirements influencing the overall architecture. "Identifying requirements" during the Macro Design phase concentrates on refining the system specifications based on all user requirements, business processes, and business requirements.

Chapter 4: Answers to questions

Here we give the answers to the questions asked in Chapter 4, “Building solutions on a multiple Business patterns basis” on page 83.

Answers to discussion and review questions in 4.6.1, “Discussion and review questions” on page 120

1. Question 1.
   - Identifying requirements: The navigation of the Patterns for e-business asset catalog starts from a broad categorization of business requirements that guide asset selection and offer insights into the business area and the main business drivers.
   - Documenting the client environment: The information found in the RFP will help document the current IT environment, which is key during the later stages of the asset selection process and at the same time creates an opportunity for a focused analysis of available documentation.
   - Defining the solution: Patterns for e-business contribute significantly to the definition of the solution.
2. Question 2.
   a. ABC is looking to provide electronic document submission capabilities to its independent agent community through the internet.

   b. While ABC Insurance expects the initial volume of electronic transactions to be small compared to its traditional business processes, they do believe that offering this infrastructure to their agent community ahead of the competition will provide growth opportunities in the future. ABC Insurance customers will see this as a speed to market opportunity, reducing the time required to process insurance applications, as well as a cost reduction strategy.

   c. The business functions provided by the solution are:
      - A personalized user/agent
      - Electronic insurance application submission
      - Integration with existing systems for insurance application processing
      - Access to insurance rating and underwriting information
      - Access to insurance application status
      - Electronic premium payment submission

3. Question 3.
   - Select the business pattern.
   - Select the application pattern.
   - Select the run time pattern.
   - Obtain product mapping.
4. The initial analysis of the high-level business requirements indicates that two Business patterns may be applicable to the solution:
   - The Self-Service Business pattern is applicable if an organization's requirements suggest that users need to interact directly with existing data. ABC Insurance is interested in providing access to its existing systems across the Web to its insurance agents.
   - The Access Integration pattern enables access from multiple channels (devices) and integrates the common services required to support a consistent user interface. This pattern is relevant to the ABC Insurance case due to the requirement for supporting the agents efficiently in their work by providing them with a single, consistent, and seamless access mechanism to the various applications.
   - The Application pattern focuses on the application level functionality required to support the identified Business patterns. Based on the available information, the following Application patterns are selected:
   - The Self-Service::Router Application pattern provides a structure for applications that require the intelligent routing of requests from multiple delivery channels to one of multiple back-end applications. ABC Insurance needs to make the functionality of a number of its existing back-end systems available across a number of different channels, initially only across a Web channel, but PDAs have been identified as a future extension.
   - The Access Integration::Web Single Sign-on Application pattern will allow access to multiple back-end systems and a consistent security policy that will be enforced across these applications. This will simplify access by ABC Insurance agents in addition to easing the administrative impact.
5. Solution Outline Phase: Defining the high level architecture activity. The main objective of this activity is to get an overall view of the structure and topology of the system to be delivered so that appropriate attention can be given to more challenging areas of the proposed solution. During this activity, the architect will validate the business and IT drivers from the client and match these against those associated with patterns. It is important that at this stage an accurate assessment be made of any gaps between the identified assets and the required solution so that adequate attention can be given to those

areas through either the selection of additional products or through custom development.

Macro Design Phase: Defining the architecture. This activity will focus on specifying the overall architecture in terms of how it addresses both the functional and non-functional requirements, so that in subsequent phases a start can be made on its implementation. During this activity, the architecture defined during the Solution Outline phase is confirmed and further refined.

6. Question 6.

   d. The focus of this activity lies in familiarizing the project team with the development method, assessing training and mentoring needs, and tailoring the development method to the needs of the project taking the use of assets into account.

   e. Using candidate assets can improve the profitability of a project as well as lower its risk given that the solution will be based on assets used in previous successful implementations.

7. The System Context Diagram is developed using the high-level descriptions of actors, the system to be delivered, and their interactions. It defines a precise definition of the scope of the system. This diagram emphasizes how the system to be delivered fits into its environment.

   The Architecture Overview Diagram (also known as a Solution Overview Diagram) gives an overview of the system that will be delivered, that is, it shows the major building blocks of the solution itself. This diagram serves as a basis for depicting all the Patterns that will be leveraged in developing a solution for an organization.

8. Question 8.

   a. Self-Service fits given that the agent is a user of ABC Insurance's business system.

   b. Collaboration fits given that the requirement for being able to access the insurance application status (an implied interactive/collaborative environment between the agent and the ABC Insurance personnel).

   c. Information Aggregation fits, as ABC Insurance is interested in providing rating and underwriting information back to the agent community.

   d. Extended Enterprise is applicable due to the need for electronic payment processing, which (in most cases) requires involving an external payment processing center like a financial institution.

9. Question 9.

   – Access Integration provides the front-end integration of multiple services and information through a common portal as requested for the ABC Insurance solution.

   – Application Integration provides for seamless back-end integration of multiple applications and data without the user accessing them directly. The new Web-based tier that will be developed to support the solution will need to access ABC Insurance's existing systems and data.

10. The Router Application pattern supports multiple delivery channels connecting one or more back-end existing applications through a router tier.

11. Question 11.

    a. The Web Single Sign-On application pattern satisfies one of ABC Insurance's requirements: to provide seamless access to multiple applications with a single sign-on while continuing to protect the security of the enterprise and its applications.

    b. The Personalized Delivery application pattern satisfies the requirement to create a personalized experience for ABC Insurance's agents. This pattern provides a framework for giving access to applications and information tailored to the interests and roles of a specific user or group.

12. The mid-tier environment for ABC Insurance will be on a single platform connecting to existing systems that exist on one platform as well, resulting in a low level of complexity. Therefore, the homogenous environment can exploit the application's security server for single sign-on functionality.

13. The activity in Solution Outline refers to the activity of determining the key requirements influencing the overall architecture. The activity in the Macro Design phase concentrates on refining the system specification based on all user requirements, business processes, and business requirements by elaborating the system requirements and specifications.

Answers to the challenge questions in 4.7.1, “Challenge questions” on page 122

1. The answer will vary, based on the date assignment made. Page 88 of the text refers students to the resource *Patterns: Custom Designs for Domino & WebSphere Integration*, SG24-6903 as a starting point for the research.

2. The answer will vary based on the research. If students conduct a Web search with the criteria "Hershey" & "SAP", they will find an abundance of information.

   Consider directing them to the following sites:

   http://www.cio.com/archive/111502/tl_hershey.html
   http://www.thespot4sap.com/articles/Rocky_Road.asp

3. The answer will vary, but look to have defenses based on the need for real time hot-swap capability for load balancing. Industries or organization

examples may include Web based, e-commerce based enterprises that need a zero fail rate, or healthcare organization that also need a zero fail rate.

Answers to team case study projects in 4.7.2, “Team case study projects” on page 122

For each case:

- Ask each team to capture the insights gained as the team collaboratively analyzed the case.
- Each team presents its analysis, solution, and insights gained to the class.
- Identify emerging themes and principles.
- With the instructor facilitating the class discussion, sharing team insights and reflection provide yet another context for deeper understanding.

Answers to multiple choice in 4.7.3, “Multiple choice” on page 127

1. D
2. A
3. A
4. B
5. C
6. A
7. D
8. C
9. D
10. B

Answers to fill in the blank in 4.7.4, “Fill in the blank” on page 133

1. Self-Service
2. WebSphere/Domino
3. Performance
4. Non-functional
5. Runtime
6. Runtime

Answers to true or false in 4.7.5, “True or false” on page 134

1. True
2. True
3. False
4. True

5. False
6. True
7. True
8. True

Answers to short answer in 4.7.6, "Short answer" on page 135

1. Question 1.

   The Self-Service business pattern is commonly observed in e-business solutions that provide users the ability to access their information and change it by interacting directly with core business systems and databases. The Collaborative business pattern occurs in e-business solutions that involve one- or two- way interactions between users of the solution.

   Examples provided by the text:

   – Customer-facing applications such as an Online Broker application, which allows investors to manage their portfolios and make equity trades across the Web, and allow investors to collaborate with the broker's help desk.

   – Business partner-facing applications, such as a dealer extranet, that allow dealers for a manufacturing company to link their sales distribution operations to their shop floor processes by exchanging forecast, demand, and production information, and also facilitate online collaboration with the manufacturer's sales department.

2. Question 2.

   – The users and customers need to directly interact with business processes.

   – The business processes need to be integrated with existing business systems and information.

   – The business process must be reachable in a common, consistent, and simplified manner through multiple delivery channels.

   – The business activity demands and fosters collaboration and the sharing of information among the participants.

3. Question 3.

   – The answer to the question is not explicitly provided in the text, but students should be able to acknowledge the impact an SAP rollout would have on operations and technical aspects of an organization. Review answers to make sure these issues, at least, are addressed.

4. Question 4.
   - Very simple implementation.
   - Very short delay for the forwarding task if the MAC forwarding method is used.
   - High availability and failover for the Web application server nodes.
   - Horizontal scalability possible by simply adding more servers to the content server load balancer cluster.
   - High availability and failover provided for the load balancer nodes.
5. Question 5.

   Advantages
   - Each node type can be configured to specific tasks.
   - A domain firewall between the two node types adds additional security.
   - More flexibility in number of nodes of each type.
   - Flexibility to not apply security on certain static content.

   Disadvantages
   - Many configuration changes on the Application Server node call for a redeployment of the plugin-cfg.xml file to all Web server redirector nodes.
   - The configuration of the domain firewall requires additional skills.

Chapter 5: Answers to questions

Here we give the answers to the questions asked in Chapter 5, "Applying Patterns with existing Custom designs" on page 137.

Answers to discussion and review questions in 5.2.3, "Discussion and review questions" on page 155

1. Question 1.

   The Self-Service Business pattern is commonly observed in e-business solutions that provide users the ability to access their information and change it by interacting directly with core business systems and databases. The Collaborative Business pattern occurs in e-business solutions that involve one- or two-way interactions between users of the solution

Examples provided by the text:

- Customer-facing applications such as an Online Broker application, which allows investors to manage their portfolios and make equity trades across the Web, and allow investors to collaborate with the broker's help desk.
- Business partner-facing applications, such as a dealer extranet, that allow dealers for a manufacturing company to link their sales distribution operations to their shop floor processes by exchanging forecast, demand, and production information, and also facilitate online collaboration with the manufacturer's sales department.

2. Question 2.

   - The users and customers need to directly interact with business processes.
   - The business processes needs to be integrated with existing business systems and information.
   - The business process must be reachable in a common, consistent, and simplified manner through multiple delivery channels.
   - The business activity demands and fosters collaboration and the sharing of information among the participants.

B

Resources

Patterns for e-Business Library Resources

http://www-128.ibm.com/developerworks/patterns

Current Patterns IBM Redbooks publications

- *Patterns: Emerging Patterns for Enterprise Grids*, SG24-6682
- *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240
- *Patterns: SOA Foundation Service Connectivity Scenario*, SG24-7228
- *Patterns: SOA Foundation - Business Process Management Scenario*, SG24-7234
- *Patterns: Building Serial and Parallel Processes for IBM WebSphere Process Server V6*, SG24-7205
- *Patterns: SOA Client - Access Integration Solutions*, SG24-6775
- *Flexible Self-Service Application Patterns Using WebSphere and Process Choreography on z/OS*, SG24-6356
- *WebSphere Application Server for iSeries V6: Building Advanced Configurations*, SG24-6637
- *Patterns: Model-Driven Development Using IBM Rational Software Architect*, SG24-7105

- *Patterns: Extended Enterprise SOA and Web Services*, SG24-7135
- *Patterns: Integrating Enterprise Service Buses in a Service-Oriented Architecture*, SG24-6773
- *Patterns: Implementing Self-Service in an SOA Environment*, SG24-6680
- *Patterns: SOA with an Enterprise Service Bus in WebSphere Application Server V6*, SG24-6494
- *WebSphere Commerce Portal V5.4 Solutions Guide*, SG24-6890
- *A Portal Composite Pattern Using WebSphere Portal V5*, SG24-6087
- *Patterns: Portal Search Custom Design*, SG24-6881
- *Using a Single Business Pattern with the Rational Unified Process (RUP)*, REDP-3877
- *Patterns: Implementing an SOA using an Enterprise Service Bus*, SG24-6346
- *Patterns: Service-Oriented Architecture and Web Services*, SG24-6303
- *Patterns: Direct Connections for Intra- and Inter-enterprise*, SG24-6933
- *Patterns: Broker Interactions for Intra- and Inter-enterprise*, SG24-6075
- *Patterns: Serial Process Flows for Intra- and Inter-enterprise*, SG24-6305
- *Patterns: Serial and Parallel Processes for Process Choreography and Workflow*, SG24-6306
- *Patterns: Information Aggregation and Data Integration with DB2 Information Integrator*, SG24-7101
- *Patterns: Flexible Self-Service Applications Using Process Choreography*, SG24-6322
- *More about High-Volume Web Sites*, SG24-6562
- *Patterns: Self-Service Application Solutions Using WebSphere for z/OS V5*, SG24-7092
- *IBM WebSphere V5 Edge of Network Patterns*, SG24-6896
- *Self-Service Applications using IBM WebSphere V5.0 and WebSphere MQ Integrator V2.1 Patterns for e-business Series*, SG24-6875
- *Patterns: Self-Service Application Solutions Using WebSphere V5.0 for iSeries*, REDP-3670
- *Patterns: Applying Pattern Approaches Patterns for e-business Series*, SG24-6805
- *Patterns: Custom Designs for Domino & WebSphere Integration*, SG24-6903
- *Patterns: Self-Service Application Solutions Using WebSphere V5.0*, SG24-6591

- *B2B e-commerce with WebSphere Commerce Business Edition V5.4 Patterns for e-business Series*, SG24-6194
- *Patterns for the Edge of Network*, SG24-6822
- *Patterns on z/OS: Connecting Self-Service Applications to the Enterprise*, SG24-6827
- *e-business Globalization Solution Design Guide: Getting Started*, SG24-6851
- *B2B e-Marketplace Patterns with WebSphere Commerce Business Edition V5.4 for the IBM eServer iSeries Server*, REDP-0212
- *Patterns: Connecting Self-Service Applications to the Enterprise using the IBM eServer iSeries Server*, REDP-0213
- *WebSphere Commerce Suite V5.1 for iSeries, Implementation and Deployment Guide*, REDP-0159
- *e-commerce Patterns for Building B2C Web Sites Using IBM WebSphere Commerce Suite V5.1*, SG24-6180
- *e-commerce Patterns for Linux on zSeries Using WebSphere Commerce Suite V5.1 Patterns for e-business series*, REDP-0411
- *e-Commerce Patterns Using WebSphere Commerce Suite, Patterns for e-business Series*, SG24-6156

Independent analyst reports

Bloor Research Analyzes the Patterns for e-business, found at:

http://www-106.ibm.com/developerworks/patterns/guidelines/bloor.pdf

The Butler Group's "Evolution of e-business Patterns" Assessment, found at:

http://www-106.ibm.com/developerworks/patterns/guidelines/butler.pdf

Component Developer Strategies review of the Patterns, found at:

ftp://www6.software.ibm.com/software/developer/patterns/cutter.pdf

White papers

Patterns for e-business - Architectural Patterns and their use in Solution Architecture and Enterprise Architecture Definition, found at:

http://www-106.ibm.com/developerworks/patterns/guidelines/arch-patterns.pdf

WebSphere and SAP Integration for Self-Service, Application Integration and Extended Enterprise, and Electronic Commerce applications, found at:

ftp://www6.software.ibm.com/software/developer/patterns/ws-sap-integration.pdf

Facilitating the application development process using the IBM Patterns for e-business, found at:

http://www-106.ibm.com/developerworks/patterns/guidelines/lord.pdf

Multimedia presentations and Screen Cams

- Introduction to the Patterns for e-business:

  http://www-106.ibm.com/developerworks/patterns/multimedia/pdk.mov
- Flash tutorial to the Patterns for e-business:

  http://www-106.ibm.com/developerworks/patterns/multimedia/version4/Flash_web.swf
- ScreenCams Presenting the newly updated Patterns Web site and solution designs:
  - ftp://www6.software.ibm.com/software/developer/patterns-zip/2001screencam1.zip (4770 KB)
  - ftp://www6.software.ibm.com/software/developer/patterns-zip/2001screencam2.zip (5940 KB)
  - ftp://www6.software.ibm.com/software/developer/patterns-zip/2001screencam3.zip (6990 KB)

WebCasts

- IBM Patterns for e-business CIO executive audio conference
  - Held November 15, 2001
  - Speakers: Srinivas Koushik, ex-IBM Distinguished Engineer and Jonathan Adams, IBM Distinguished Engineer
  - Download a copy of conference material at the following URLs:
    - ftp://www6.software.ibm.com/software/developer/patterns/cio.pdf
    - ftp://www6.software.ibm.com/software/developer/patterns/cio.zip

Patterns for e-business Visio Stencil

Use the following link to download the Patterns for e-business Microsoft® Visio® stencil and to get instructions how to deploy the file. Thanks are due to David Janson (dhjanson@us.ibm.com) for providing the stencil:

http://www-128.ibm.com/developerworks/patterns/library/visio.html

Chapter 1

- *Patterns of Experience: A Review of IBM Patterns for e-Business Initiative*, found at:

  http://www-128.ibm.com/developerworks/patterns/library/index.html
- *Patterns for e-business – continued evolution*, found at:

  http://www-128.ibm.com/developerworks/patterns/library/index.html
- *Component Development Strategies Monthly Newsletter*, found at:

  http://www-128.ibm.com/developerworks/patterns/library/index.html
- *Facilitating the application development process using the IBM Patterns for e-business*, found at:

  http://www-128.ibm.com/developerworks/patterns/library/index.html

Glossary

Actor A term from Use Case modelling and UML defined by the Object Modelling Group. It is anything, human or otherwise, that interacts with a Business Function.

Application and Runtime pattern alignment Application and Runtime pattern alignment displays the logical alignment between the application tiers and the supporting (middleware) nodes.

Application Architecture (Functional Architecture) Focuses on the application portfolio required to support the mission and information needs of the organization. At the next level of detail, it addresses the common business components and business services that can be leveraged by multiple applications.

Application data This represents a logical split of the application data with or without metadata.

Application pattern Application patterns represent the logical partitioning of the application logic and data together with the styles of interaction between the logic tiers.

Application Programming Interface (API) Defines how programmers use a particular feature provided by an application. The As-Is Host Application pattern provides wider intranet access to existing host applications.

Application tier A tier is a logical layer within a design that allows the subdivision of the application into major functional collections. Tiers can be further subdivided into functional components.

Authentication The process of determining whether someone or something is, in fact, who or what it is declared to be. In private and public computer networks (including the internet), authentication is commonly done through the use of sign-on passwords. Knowledge of the password is assumed to guarantee that the user is authentic.

Authorization The process of giving someone permission to do or have something. A system administrator defines which users are allowed access to the system and what use privileges they are given (such as access to which file directories, hours of access, and amount of allocated storage space). Logically, authorization is preceded by authentication.

B2B gateway A B2B gateway manages the interactions between the trading partners within an e-Marketplace based on an executable contract. An example of an executable contract is a Trading Partner Agreement (TPA), which is an XML document that defines the general ground rules for such functions as pricing quotations, orders, and acceptances. It also describes the format for the communications that will be used and provides specifics on actions, security, and error handling. When referring to the actions within a TPA, the trading partner is required to implement service interfaces corresponding to these actions. The service interfaces are started by an executable version of the TPA and executed under the application server. Some service implementations require message brokering services to interact with enterprise applications. This can be provided by the Business Protocol Manager or delegated to a dedicated message brokering server.

Batch Batch is an offline processing environment, usually running at night or on the weekend, where long-running and resource-intensive jobs are run. In a data warehouse it is used extensively for data acquisition and propagation.

BDW (business data warehouse) A business data warehouse (BDW) is a data store containing detailed, reconciled, and historical basic business data, structured according to an enterprise data model and designed to be the single, consistent source for all information required for business intelligence purposes. It is guaranteed to be integrated and consistent across the breadth of the business and to cover the span of required history of the business. The business data warehouse is seldom accessed directly by users and then solely in read-only mode.

Best practices The design development, deployment and management guidelines for building an application based on a closed set of nested patters. These guidelines may be generic to a family of middleware products (with some variations), generic to an industry standard set of technologies (with some vendor variations), or specific to a product or product family

BI application client A BI application client is the process, running on a user workstation, that provides business intelligence function to the user. Operating in client-server mode, it works with either a BI app server, BDW server, or ODS server to provide access to data.

BI application server A BI application server is the process, running on an internal node that provides access to data residing in a data mart. Depending on how function is split between the BI app client and the server, the BI app server may range in complexity from simply making database calls to high-function BI processing like online analytical processing (OLAP).

Binding Application Binding applications are used to join B2B interactions with suppliers and buyers to the marketplace commerce functions. They would use a B2B protocol and message formats, which generally are XML messages, like cXML (Commerce XML), to communicate with partners. Operations such as performing an inventory check or obtaining fulfilment information would be initiated by the Commerce and Web app server and flow through these applications out to the partners. Similarly, programmatic requests would flow into the marketplace through these applications, which would interact with the Commerce and Web app server.

Broker application pattern Occurs in designs where one application can access other applications through a set of business services that offer a large degree of independence between the initiating application.

Browser A browser is a workstation-based tool, such as Netscape or Internet Explorer®, for accessing the Internet and other resources

Business architecture Addresses the business mission, strategy, line of businesses, organization structure, business process models, business functions, and so on.

Business Flow Manager A Business Flow Manager controls the invocation and interaction of the applications implementing a (major) task in the B2B workflow. In some cases, it simply “is” the business application; interpreting messages and invoking the appropriate back-end system services, potentially through a message router. In more sophisticated implementations, XML scripting might be used to knit together the various application components needed to complete the a task and allow easy extension and modification of this “micro workflow' implementing the current business process step. The Business Flow Manager controls the business processes used within an organization and the points at which a trading partner's processing must be integrated with these processes.

Business Intelligence (BI) Business Intelligence (BI) is the gathering, management, and analysis of large amounts of data in order to gain insights to drive strategic business decisions, and to support Operational processes with new functions. BI is about the development of information that is conclusive, fact based, and actionable. It includes technology practices like data warehouses, data marts, data mining, text mining, and online analytical processing (OLAP). The objective of a BI solution is to transform data into useful information, such as customer profiles, buying habits, product profitability, and competitive analysis. It may involve analyzing volumes of data for unsuspected, but valuable, associations and insight. It includes streamlining data into useful reports and sharing that information with people inside and outside the organization who need that information.

Business pattern Business patterns highlight the most commonly observed interactions between Users, Businesses, and Data. They are the fundamental building blocks of most e-business solutions, and describe the interaction between the participants in an e-business solution. The four Business patterns are:

- Self-Service
- Collaboration
- Information Aggregation
- Extended Enterprise

Business-to-Business Business-to-Business (now know as the Extended Enterprise Business pattern) describes programmatic links to both intra-business and inter-business, for example, electronic invoicing.

Cascading Style Sheets (CSS) A simple mechanism for adding a presentation style to Web documents.

CICS The Customer Information Control System is the world's leading online transaction monitor.

Client The requesting program in a client/server model. For example, a browser is a client that requests services from a Web server.

Coarse grained application (CGA) A collection of major Business Functions identified from the Business Requirements Specification as forming a logical grouping that can be dealt within the patterns identification process as a single unit.

Collaboration business pattern Can be observed in e-business solutions that allow users to communicate and share data and information with other users or groups of users in the network.

Collaboration Doing business is a series of Collaboration processes. It requires interaction between employees, vendors, suppliers, and Business Partners. While e-mail is one example of an indispensable communication tool used by companies around the world, a number of other collaborative applications are increasingly coming into play. These applications enable local workgroups, or even geographically dispersed teams, to work together using real-time information sharing and distribution across the internet. The applications include e-mail, group calendaring and scheduling, shared document libraries, discussion databases, newsgroups, and so on.

Compensating transactions Used to restore data to a state that existed before the action causing the change (thus reversing the update).

Composite pattern Composite patterns combine Business patterns and Integration patterns to create complex, advanced e-business applications. Of course, there are numerous potential combinations of Business patterns and Integration patterns, but only a limited number of Composite patterns documented. A solution design composed of these multiple building blocks is only considered a Composite pattern when it is recurrently employed to solve the problems of businesses across a wide range of industries. Examples of such Composite patterns include:

- Electronic Commerce
- e-Marketplace
- Portals
- Account Access

Connection A connection represents the network infrastructure over which interactions between application tiers placed on separate nodes take place. Connections may "carry" one or more interactions, and there may be more than one connection between any two nodes.

Content management architecture Identifies what type of content is required, and how is it maintained, published, and distributed.

Core System An application system regarded by an enterprise as vital to its business survival.

Customer Relationship Management (CRM) Customer Relationship Management (CRM) generally means any effort undertaken to improve customer service. Our definition encompasses a broader vision, based on new approaches for managing customer relationships for competitive advantage and differentiation. Enabled by e-business technology, this approach focuses on managing the customer "holistically," so you can:

- Provide consistent, value-add experiences through a single, long-running dialog.
- Deal with each customer as an individual.
- Recall and reference all previous interactions.
- Accommodate each customer's needs and preferences.

Customized Presentation to Host application pattern Used to provide a more user-friendly interface to existing host applications without changing the underlying application.

Data Data represents a logical split of the application data with/without metadata.

Data cleansing Data cleansing is a type of data transformation specialized to detect and remove a number of errors, including format errors and content errors, from data during data acquisition.

Data mart A data mart is a data store defined and designed to meet the information needs of a department or group of users. It contains needed data, detailed or summarized, and preferably sourced from the business data warehouse. Data marts are the primary sources of information for users, are optimized to satisfy their query or reporting needs, and are usually used in read-only mode.

Data mining Data mining is a method of searching for unexpected patterns and relationships in data using a variety of mathematical techniques and tools.

Data Mirroring Application pattern Enables integration of disparate applications by making a read-only copy of the data available to a partner application.

Data propagation Data propagation includes the processes, tools, and services responsible for the derivation of data from the business data warehouse into data marts, including the capture or extraction of data from the BDW, its transformation according to predefined rules, and its load or application to the data mart.

Data-tier connection A data-tier connection represents the placement of data relative to the application tiers.

Data transformation Data transformation includes any processes used in data acquisition or data propagation that convert data from one form to another by the application of rules, lookup tables, or combination with other data.

Data warehouse services Data warehouse services include the processes, tools, and services needed to manage and run the data warehouse environment, including archive/restore, security, database management, process management, and so on, but excluding data acquisition and population.

Data warehouse A generic term usually used to cover all of the components involved in the provision of business intelligence, from data acquisition to the data marts, as well as the metadata and services elements of business intelligence. It can also be used as a synonym for business data warehouse, or to represent a simple unlayered data warehouse structure.

DB2 (Database2) The implementation by IBM of the relational database model.

DB2 remote Local DB2 system that forwards a database (db) request using distributed relational database architecture (DRDA®) to the remote DB2 system for execution.

Decision support system (DSS) The decision support system (DSS) is one of a number of older synonyms for applications and data used to support decision-making and business management processes, now broadly called business intelligence systems

Decomposition application pattern Extends the hub-and-spoke architecture provided by the Router Application pattern. Decomposes a single, compound request from a client in several simple requests and intelligently routes them to multiple back-end applications. Typically, the responses from these multiple back-end applications are recomposed into a single response and sent back to the client.

Delivery Manager The delivery manager is responsible for routing of information to applications that implement the business processes managed by the business flow manager. It helps to structure a system design that allows a pair of applications to directly communicate with each other.

Directed Collaboration Application pattern Allows users to collaborate with others in the network interactively.

Directly Integrated Single Channel Application pattern Provides a structure for applications that need one or more point-to-point connections with back-end applications, but only need to focus on one delivery channel.

Distributed Component Object Model (DCOM) A set of Microsoft concepts and program interfaces in which a client program object can request services from server program objects on other computers in a network.

Device Support service Enables users of a wide range of devices to access the same set of applications.

Digital certificates A digital certificate is an electronic credential issued by a trustworthy organization such as a bank, credit union, or large company. The digital certificate vouches for an individual's, business', or organization's identity and authority to conduct any transaction over the internet. The issuing organization is called a Certificate Authority (CA). VeriSign is an example of an existing CA.

Digital certificates address the issue of ensuring that the owner of a public key is really who he claims to be. This technology provides a mechanism to distribute public keys in a special format called a certificate. In addition to the key itself, the certificate contains information about the sender. The whole package is then signed by the issuing CA using its private key. The receiver of the certificate can then verify the CA's digital signature using the CA's public key. Digital certificates are made available to the public through online directories based on X.500 standards.

Digital certificates are defined by the X.509 standard. A X.509 certificate is typically a small file that contains:

- Subject's distinguished name
- Issuer's distinguished name
- Subject's public key
- Issuer's digital signature
- Validity period
- Certificate's serial number

Direct Connection Application pattern Helps to structure a system design that allows a pair of applications to directly communicate with each other.

Directed Collaboration Application pattern Allows users to collaborate with others in the network interactively.

Directly Integrated Single Channel Application pattern Provides a structure for applications that need one or more point-to-point connections with back-end applications, but only need to focus on one delivery channel.

Distributed Component Object Model (DCOM) A set of Microsoft concepts and program interfaces in which a client program object can request services from server program objects on other computers in a network.

Document Exchange application pattern Helps to structure the batched electronic exchange of data using mutually agreed message formats.

Domain Name System (DNS) The DNS assists in determining the physical network address associated with the symbolic address (URL) of the requested information. The Domain Name Server Node provides the technology platform to provide host to IP address mapping, that is, to allow for the translation of names (referred to as URLs) into IP addresses and vice versa.

e-business solutions Use evolutionary technology and re-engineered business processes to develop revolutionary new applications that are not limited by time, space, organizational boundaries, or territorial borders.

EDI translation package In its simplest form, an Electronic Data Interchange (EDI) translator converts EDI transaction sets (EDI messages) to and from flat files, into a usable format for an enterprise's applications. The translator can read batches of messages from a VAN mailbox and process them. More sophisticated translation packages convert the message to a request to a transaction processing system.
The EDI translator is responsible for mapping to and from EDI standard formats such as X12 or EDIFACT and the communications with the VAN mailbox for sending and receiving documents.

Enterprise Solution Structure (ESS) Enterprise Solutions Structure (ESS) is a major IBM initiative to establish a standard architectural framework to support creation, reuse, and maintenance of architecture and design assets. These intellectual capital assets are used by IBM services practitioners for developing and delivering enterprise solutions. ESS draws on experiences with building customer solutions to distill "best practice" structures, models, and sample deliverables. The framework provides a rich set of architectural building blocks for solution architects and provides guidance on when and how to use this content to advantage. Specifically, this architecture provides a common, consistent approach for understanding and documenting business requirements through a business model, designing a logical architecture of key components and services, and finally, implementing a physical architecture based on actual products, platforms, and services.
The term "reference architecture" is used to refer to the collection of assets, which as a whole describe how to implement a given type of business solution. For example, there is a reference architecture, which shows how to implement a call center. There is another one, which shows how to implement an online buying application.

Electronic commerce solutions Allows enterprises to offer products and services to existing and new customers across new channels based on internet technologies.

Electronic data interchange (EDI) A standard format for exchanging business data. An EDI message contains a string of data elements, each of which represents a singular fact, such as a price, product model number, and so forth, separated by a delimiter. The entire string is called a data segment. One or more data segments framed by a header and trailer form an EDI transaction set.

e-Marketplace An e-Marketplace consists of a hub that brings multiple buyers together with multiple sellers. Marketplaces provide value to their members by providing a unified view of the set of goods and services traded in the market and by providing a variety of mechanisms to facilitate trade in such products. Many marketplaces feature an aggregated catalog containing product information from multiple suppliers. This catalog can be presented to the user with different views based on the role under which the user accesses the marketplace. Organizations are able to define roles, with authentication, and an approval workflow based upon these roles.

Emerging pattern This pattern is considered an emerging pattern, meaning it is based on newer technology that has been tested in the lab, but not necessarily in a commercial environment. Emerging patterns are useful in a situation where you need to use newer technology to solve concerns that proven patterns might not address.

Enterprise data model An enterprise data model is a consistent definition of all the data common to the business, from a high-level business view to a generic logical data design, including links to the physical data designs of individual applications.

Enterprise resource planning (ERP) A full implementation of enterprise resource planning implies a top-to-bottom transformation of the way a company operates, does business, and faces the future. Changes are set in motion in how the company is managed, how it serves customers, and how it reacts to the marketplace. The potential benefits are greater customer satisfaction, reduced cycle time, and increased productivity.
ERP-packaged software covers the primary back office functions of a business. ERP solutions encompass the ERP software plus hardware, database, integration software, technical processes, and people. In this wide perspective, a new information technology infrastructure is part of a large moving picture that can involve reengineering business processes, overhauling the roles, responsibilities, and daily routines of personnel, and implementing application software.

Enterprise-Out Used to represent both an attitude of mind (enterprise-centric) and the consequent extension of existing applications out to a new Web-based access channel.
The mindset here is that there is a valuable existing system in place. The company simply wants add a new access channel to enable users to do business with that system (enhanced as needed) from the Web.
In this case, as there will be very close integration with and re-use of existing back-end systems, the approach is often referred to as an Enterprise-Out mindset.

ESCON® Channel A communications method between an S/390® server and other machines.

Executive information system (EIS) Provides high-level business information (balanced scorecard figures, key indicators, and so on) to upper management or executives. This information is often delivered to the user as a discrete item of work either automatically or on-demand.

Exposed Application Application pattern Helps to structure a system design that allows specific applications to be directly accessed by partner systems across organizational boundaries.

Exposed Business Services application pattern Structures a system design that exposes specific services that can be directly invoked by partner systems across organizational boundaries.

Extended Enterprise Business pattern Addresses the interactions and collaborations between business processes in separate enterprises.

Extensible Markup Language (XML) A universal format for defining structured documents and data on the Web.

Extensible Stylesheet Language (XSL) A language for creating style sheets. It describes how XML data is to be transformed and formatted before it is presented to a user or passed on to another system.

Extract + (transform) + (load) Also known as ETL. A combination in a single process of extract function, transform function, and any load function required.

Extract + (transform) A combination in a single process of extract function and any transform function required.

Extract The process of reading data from a source database. The data may be extracted as a complete set, a partial set, or simply the changes that have occurred over a period of time. Extract from the operational systems is the first phase of data acquisition when populating the BDW, and the first phase of data propagation when populating a data mart from the BDW.

Firewall A hardware/software system that manages the flow of information between the internet and an organization's private network. Firewalls can prevent unauthorized internet users from accessing private networks connected to the internet, especially intranets, and can block some virus attacks, as long as those viruses are coming from the Internet. A firewall can separate two or more parts of a local network to control data exchange between departments. Components of firewalls include filters or screens, each of which controls transmission of certain classes of traffic. Firewalls provide the first line of defence for protecting private information, but comprehensive security systems combine firewalls with encryption and other complementary services, such as content filtering and intrusion detection.
Firewalls control access from a less trusted network to a more trusted network. Traditional implementations of firewall services include:

- Screening routers, (the Protocol Firewall)
- Application gateways (The Domain Firewall)

A pair of Firewall Nodes provides increasing levels of protection at the expense of increasing computing resource requirements. The Protocol Firewall is typically implemented as an IP Router.

Guideline A directing or standardizing principle laid down as a guide to procedure, policy, and so on.

GUI Graphical User Interface.

Hardware The physical components of a computer system.

HDML Handheld Device Markup Language is used to format content for Web-enabled mobile phones. HDML is phone.com's (formerly known as Unwired Planet) proprietary language, which can only be viewed on mobile phones that use phone.com browsers. HDML came before the WAP standard was created. It uses phone.com's Handheld Device Transport Protocol (HDTP), instead of WAP. Phones access HDML sites the following way:
Once the URL is typed into the phone, the phone sends the request to phone.com's UP.Link gateway. The gateway sends a HTTP request to the Web server. The Web server returns the page through HTTP back to the phone.com UP.Link gateway. The gateway sends the data through HDTP to the wireless carrier's network and down to the phone. HDML and the phone.com gateway are most popular throughout North America. In Europe, WML and the Nokia WAP gateway and browser are the emerging standard. However, some versions of phone.com browsers do interpret basic WML. To serve HDML pages, administrators must add the text/x-hdml mime type to their Web servers.

HOLAP (Hybrid OLAP) As the name suggests, this is software that uses both ROLAP and MOLAP techniques together to enable multi dimensional representation of data.

HTML (Hyper-Text Markup Language) A computer language used to write and format pages for Web publication. Incorporates Web features such as hyperlinks, frames, headings, and so on.

HTTP (Hyper-Text Transfer Protocol) The underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

HTTPS Hyper-Text Transfer Protocol Secure.

IBM Global Services IBM Global Services refers to that organization within IBM that provides professional IT Services and Consulting.

IBM Universal Server Farm A set of outsourcing sites run by IBM. Server technology from multiple different vendors is operated by IBM on behalf of organizations that would rather not manage their own server operations. Frequently in Online Buying sites, the nodes in the DMZ front end and the Data Base Server Node are outsourced. These are then networked to the company's own datacenter for back-end order processing.

IDE Integrated development environment.

IDL Interface Definition Language, the language used to define CORBA interfaces.

Identification The process by which an IT system determines which person or computer process is associated with a request to use IT system resources.

IIOP (Inter-ORB Inter-Operability Protocol) The protocol used by CORBA applications.

i-mode The packet-based service for mobile phones offered by Japan's leader in wireless technology, NTT DoCoMo. Unlike most of the key players in the wireless arena, i-mode eschews WAP and uses a simplified version of HTML, and it uses Compact Wireless Markup Language (CWML) instead of WML.

Information Access-Read-Only Application pattern Helps structure a system design that provides read-only access to aggregated information.

Information Aggregation Business pattern Can be observed in e-business solutions that allow users to access and manipulate data aggregated from multiple sources.

Informational environment The informational environment is the systems environment that supports business intelligence and in which the business data warehouse, data marts, and supporting infrastructure are built.

Information Architecture (Data Architecture) Defines what information needs to be made available to accomplish the mission, to whom, and how.

Information catalog Includes the logically single, but physically distributed metadata store of the data warehouse, containing all metadata required by business intelligence users in a form suitable for them, as well as the tools to practically use this metadata. The information catalog can also be used by developers and administrators of the warehouse.

Information portal An information portal is a user environment on the internet or intranet bringing together all the tools and data required by a user. It is customizable to appear according to the individual user's tastes and preferences.

Integration architecture Documents the recommended integration approach for front-end and back-end application integration.

Integration pattern Complex e-business applications can be built by combining multiple Business patterns together. This is accomplished by using Integration patterns as the "glue" between Business patterns. Integration patterns are differentiated from Business patterns in that they do not themselves automate specific business problems. Rather, they are used within Business patterns to support more advanced functions, or to make Composite patterns feasible by allowing the integration of two or more Business patterns. The two Integration patterns are:

- Access Integration
- Application Integration

Intelligent agent An advanced software system that is capable of performing tasks on behalf of a user in accordance with predefined roles. This may include selective information retrieval or easier navigation; problem solving, for example, finding faults in networks; and tailoring information for users through multimedia presentation.

Intermediate data Any temporary data store used by the extract, transform, or load processes.

Internet (or the Net) A global inter-network of computer networks, connected through the Internet Protocol (IP) and the world's telecommunications infrastructure. IP enables applications such as e-mail, the WWW, file transfer, and other services to run across different networks and operating systems.
As of 1999, the internet has more than 200 million users worldwide, and that number is growing rapidly. More than 100 countries are linked into exchanges of data, news, and opinions. Unlike online services, which are centrally controlled, the internet is decentralized by design. Each Internet computer, called a host, is independent. Its operators can choose which internet services to use and which local services to make available to the global Internet community.
There are a variety of ways to access the internet. Most online services, such as ibm.com, offer access to some Internet services. It is also possible to gain access through a commercial Internet Service Provider (ISP).

Intranet An intranet is a private network that is contained within an enterprise. It may consist of many interlinked local area networks and also use leased lines in the wide area network. Typically, an intranet includes connections through one or more gateway computers to the outside internet. The main purpose of an intranet is to share company information and computing resources among employees. An intranet can also be used to facilitate working in groups and for teleconferences. An intranet uses TCP/IP, HTTP, and other Internet protocols and in general looks like a private version of the internet. With tunnelling, companies can send private messages through the public network, using the public network with special encryption/decryption and other security safeguards to connect one part of their intranet to another. Typically, larger enterprises allow users within their intranet to access the public internet through firewall servers that have the ability to screen messages in both directions so that company security is maintained. When part of an intranet is made accessible to customers, partners, suppliers, or others outside the company, that part becomes part of an extranet.

IP (Internet Protocol) Internet Protocol RFC 1812. Part of the TCP/IP family of protocols describing software that tracks internet addresses, directs outgoing messages, and recognizes incoming messages. The current version is IPv4; the next version will be IPv6. It is used in gateways to connect networks at a high level.

IP address A unique set of digits (for example, 123.123.12.1) identifying a computer connected to a network, and used by communications programs. Human users communicate through more friendly domain names (for example, person@company.com.au), which are automatically translated into the IP address.

Defined in the dictionary as - To utter or do a second time or many times; to repeat. Refers to a Step or series of Steps in the Patterns identification process that need to be repeated.

Java Foundation Classes See JFC.

Java IDL A Java core API providing a CORBA/IIOP interface from Java. It complies with OMG standard Java to OMG IDL mapping. See `http://java.sun.com/products/jdk/idl/` for more information.

Java Message Service See **JMS**.

Java Naming and Directory Interface™ See **JNDI**

Java ORB An ORB that has been customized to be used from Java, that is, it has Java specific bindings (CORBA does not support Java bindings). Also applies to an ORB that can be downloaded as an applet and then used for subsequent C-S interactions.

Java servlets See **Servlets.**

Java Transaction API See **JTS.**

JavaScript™ A browser scripting language originally developed by Netscape and now standardized as ECMAscript by the European Computer Manufacturers' Association (ECMA). The Microsoft implementation is known as Jscript. Even though it contains the word "Java", it has little resemblance to Java and has a very different purpose and use. Server-side implementations are also available (for example, from Netscape).

JCE Java Cryptographic Extension; a Java extension to provide cryptographic services, such as certificate generation and message signing, for application developers. Sun™ JCE is not exportable outside the US. Australian and other implementations are available.

JCL Job Control Language.

JDBC™ The Java Data Base Connectivity (JDBC) standard provides a standard Java API to access leading vendor databases using JDBC drivers supplied by vendors or third parties. See `http://java.sun.com/jdbc` for more information.

JDK™ Java Development Kit (JDK) consists of the base reference development environment provided by Sun for developing Java applets and applications. It provides a simple Java development environment. For a real-life project, commercial Java IDEs may be used. See `http://java.sun.com/jdk` for more information.

JFC Java Foundation Classes (JFC) are second generation GUI Java libraries implementing a consistent look and feel on all operating platforms. They consist of Swing, Pluggable look and feel, and other libraries. See `http://java.sun.com/products/jfc/` for more information.

JMS Java Messaging Services provides a standard API for accessing and providing asynchronous messaging facilities to applications. Leading messaging vendors such as IBM MQSeries®, TIBCO, supports it. See `http://java.sun.com/products/jms/` for more information.

JNDI A standard Java extension API to access and provide directory and naming services. Provides a supplier API using a multitude of directory services such as X500, LDAP, CORBA Naming, DCE naming, Novell Directory Service (NDS). Also provides an accessor API for applications to use the naming and directory services.

JTS Java Transaction Service specifies the implementation of a Transaction Manager, which supports the Java Transaction API. JTS uses the standard CORBA ORB/TS interfaces and internet Inter-ORB protocol (IIOP) for transaction context propagation between JTS Transaction Managers.

Knowledge management Knowledge management includes the processes, methodologies, and tools needed to gather, manage, and effectively use knowledge in an organization. Knowledge in this context is defined as a mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information (L. Prusak).

LDAP (Lightweight Directory Access Protocol) Refers to the protocol that is used to communicate from a calling program (running on a node such as a Commerce Server) and a Directory Node. Information is kept on the LDAP-based directory node about such topics as people or services. For example, the directory could be used to store information needed to identify registered shoppers (referred to as authentication). It could also be used to store information about what functions those shoppers are allowed to perform after being identified (referred to as authorization).

Load The process of writing data into a target database. The target database may or may not previously exist, and the data to be loaded may be a complete set or simply the changes to be made to the target. Load is the final phase of data acquisition when populating the BDW, and the final phase of data propagation when populating a data mart.

Load + (extract) A combination in a single process of load and any extract function required.

Logical node A node represents a logical collection of application and middleware functionality. A node is a place where an engineered combination of application and middleware components and data will be placed.

Logical pattern A set of conceptual layouts composed of Application patterns and Runtime patterns. The following are logical patterns:

- Application patterns: Describe how the application logic and data are partitioned and how they interact.
- Runtime patterns: Uses nodes to group functional requirements. The nodes are interconnected to solve a business problem. The choice of Application pattern will typically lead to an underpinning Runtime pattern.

Mail server A computer in a network that provides "post office" facilities. It stores incoming mail for distribution to users and forwards outgoing mail through the appropriate channel. The term may refer to just the software that performs this service, which can reside on a machine with other services

Major business function Functions identified in the Business Requirements Specification as being of primary significance, for example, Demand Planning and Auctions.

Managed Collaboration Application pattern Builds on the Directed Collaboration pattern by implementing workflow rules to manage the collaboration between users of the solution.

Managed Process appllcation pattern Provides support for long-running transactions (or conversations) that underpin many business processes.

Managed Public and Private Processes Application pattern Structures a system design that handles different business protocols with different business partners and maps long-running external transactions to internal business processes and workflow.

Managed Public Processes Application pattern Structures a system design that handles the management of shared business processes between business partners.

Message oriented middleware (MOM) Provides program-to-program communications by message passing. It provides an infrastructure that supports reliable and scalable high-performance distributed application networks.

Management information system (MIS) Management information system (MIS) is one of a number of older synonyms for applications and data used to support decision-making and business management processes, now broadly called business intelligence systems

Mandatory Business pattern A Business pattern that forms one of the set that are all required in order for an identified set of patterns to match the definition of a Known Composite pattern.

Message broker A message broker is built on a queue manager and routes messages to applications. A message broker can provide real-time, intelligent rules-based message routing and dynamic message-content transformation and formatting. In this runtime, the message broker allows multiple applications to implement a published service with the broker providing application integration.

Metadata Information that describes the meaning and structure of business data, as well as how it is created, accessed, and used.

Metadata warehouse management Metadata and warehouse management are the set of functions, sometimes combined in a single product, that manage and store metadata and provide basic management functions of a warehouse's infrastructure, such as data acquisition and data propagation.

MQSeries Provides heterogeneous any-to-any connectivity from desktop to mainframe.

MQSeries Integrator Combines a one-to-many connectivity model, plus transformation, intelligent routing, and information flow modeling. It facilitates the development of new application services that integrate the functions of multiple, disparate existing business systems. More information can be found at `http://www.ibm.com/software/ts/mqseries/integrator/`.

MQSeries Workflow Business process management software from IBM. It facilitates the rapid development and management of the business processes that integrate the IT and organizational infrastructure of a company. More information can be found at `http://www.ibm.com/software/ts/mqseries/workflow/`.

Node A node represents a logical collection of application and middleware functionality. A node is a place where an engineered combination of application and middleware components and data will be placed.

ODS (operational data store) An operational data store is a data store containing detailed, partially reconciled, and nearly current data used for immediate reporting needs and sometimes also written to by users.

Online analytical processing (OLAP) Online analytical processing is a type of analysis of business data characterized by real-time consolidation and drill-down along many dimensions of large volumes of detailed data, conceptually structured in a multi-dimensional cube.

Open Buying on the Internet (OBI) A proposed standard for business-to-business purchasing on the internet, aimed particularly at high-volume, low-cost-per-item transactions. More information can be found at `http://www.openbuy.org`.

Operational architecture .Defines the software and hardware nodes necessary to satisfy the Qualities of Services requirements.

Operational data Operational data is the data managed and controlled by an operational system, and used to run the business at a detailed level, on a second-by-second basis. Such data is usually generated internally, but now often comes from an external source.

Operational Data Store Application pattern Allows an application to access a consolidation of multiple data sources, where each source may be updated and controlled by a separate application.

Operational environment The operational environment is the systems environment that supports the day-to-day operations of running the business and in which the operational systems are built.

Operational system One of the online transaction processing (OLTP) systems of the business that gather and manage the basic business data needed to run the business on a second-to-second basis. The operational system might be an existing system, a modern internet-enabled application, or anything in between.

Packages (COTS) COTS (commercial off-the-shelf) describes ready-made products that can easily be obtained. The term is sometimes used in military procurement specifications.

Parallel Sysplex® Refers to the ability of an IBM S/390 hardware environment running MVS™ to operate multiple copies of the system platforms as through they were one large single image

Partner Middleware Node One of the central attributes of this approach to business-to-business integration is that the only requirements placed on the partner is to follow the agreements on communications, security, and business protocol, including message formats.
The middleware used by the partner is not prescribed in any way and the solution is not dependent on it. This node is a placeholder for any middleware the partner chooses to deploy to assist in implementing the agreements on communications, security, and business protocol. Nevertheless, there are benefits when both partners deploy the same B2B integration nodes, such as automated configuration and version management, and fully synchronize public process execution and recovery.

Pattern A model for building and a design to copy.

Personalization The personalization functions define the roles, organizations, and individual players in the e-Marketplace. These roles imply actions that can be performed and data that can be seen. An example of personalization is defining different views of the catalog for different members.

Pervasive Device Access Application pattern Provides a structure for extending the reach of individual applications from browsers and fat clients to pervasive devices such as PDAs and mobile phones.

Physical pattern A selection of products that map to a given logical pattern.

Point-to-Point Application pattern Allows users to directly address other users in the network using simple point-to-point synchronous communications, and then enables the users to begin a direct communication link.

Population Crawling and Discovery Application pattern-Provides a structure for applications that retrieve and parse documents, and then create an index of relevant documents that match the specified selection criteria.

Population-Multi-Step Application pattern Structures the population of a data-store with structured data that requires extensive reconciliation, transformation, and restructuring.

Population-Single-Step Application pattern Structures the population of a data-store with data that requires minimal transformation and restructuring.

Population-Summarization Application pattern Extends the capabilities provided by the Population-Crawling and Discovery Application pattern by attaching summary information to index entries.

Portal solution Aggregates multiple information sources and applications to provide a single, seamless, personalized access for its users.

Primary Business patterns The four Business patterns identified as the fundamental building blocks of most e-business applications.
They are Self Service, Collaboration, Information Aggregation and Extended Enterprise.

Product mapping The Product mapping documents all the software platforms and middleware products, but the detailed configurations necessary to deliver the non-functional requirements are not defined.

Proxy server Also called a “proxy” or “application level gateway,” it is an application that breaks the connection between sender and receiver. All input is forwarded out a different port, closing a straight path between two networks and preventing unauthorized persons from obtaining internal addresses and details of a private network.
Proxy servers are available for common internet services; for example, an HTTP proxy is used for Web access, and an SMTP proxy is used for e-mail. Proxies generally employ network address translation (NAT), which presents one organization-wide IP address to the internet. It funnels all user requests to the internet and fans responses back out to the appropriate users. Proxies may also cache Web pages, so that the next request can be obtained locally. Proxies are only one tool that can be used to build a firewall.

Programmatic link Connection between two or more applications allowing the applications to interoperate.

Prune The removal of superfluous Business and Integration patterns from the final stages of the pattern identification process.

Public key infrastructure (PKI) A system for verifying the authenticity of each party involved in an internet transaction, protecting against fraud or sabotage, and for nonrepudiation purposes to help consumers and retailers protect themselves against denial of transactions. Trusted third-party organizations called Certificate Authorities issue digital certificates, attachments to electronic messages that specify key components of the user's identity. During an internet transaction, signed, encrypted messages are automatically routed to the certificate authority, where the certificates are verified before the transaction can proceed. PKI can be embedded in software applications, or offered as a service or a product. e-business leaders agree that PKIs are critical for transaction security and integrity, and the software industry is moving to adopt open standards for their use.

Purchaser node In the e-Marketplace, the purchaser node represents a personal computer that supports a commercial Web browser. It is increasingly common for this node to also be a pervasive computing device capable of displaying content delivered by the e-Marketplace.
The level of browser is expected to support SSL and some level of DHTML. Most online buying applications send a "cookie" to the browser on this node to maintain the shopping session. The cookie contains the session ID, which is used to re-establish the conversation between each of the user's interactions with the online buying program.

Queue manager Messages are sent to and received from queues that are managed by a queue manager. A queue manager provides a persistent message store and additional services including transaction support and routing of messages to the proper queue. The receiver of a message can be an adapter that transforms the message data into parameters to use on method or procedure calls to a non-queue-based application. Similarly, application adapters can convert information returned from a procedure or method call into a message, which is then sent back to the originator of the request message.

Replicated database A copy of a database where changes to both the original and replicated copy have been synchronized so that each contain the same information.

Retail customer node A personal computing device, such as a PC, supporting a commercial browser (for example Netscape Navigator or Internet Explorer). The level of the browser is expected to support SSL and some level of DHTML ”cookies” to the browser on this node in order to maintain the shopping session. The cookie will contain a session ID, which can be used to reconnect with a partially filled shopping basket or order and to re-establish the conversation for each interaction.

Router node Used in an application topology to represent the node that links presentation nodes to application nodes. It provides a common interface to multiple back-end applications, facilitating the connection, but containing no business logic to combine the application data.

Runtime pattern Defines the logical middleware structure needed to support an Application pattern. It depicts the major middleware nodes, their roles, and the connections between these nodes.

Search engine node The search engine node responds to user requests to search certain areas, such as the catalog of the e-Marketplace.

Security architecture Focuses on various aspects of the security and privacy requirements.

Security node This node is a logical representation of the functions needed to manage the security of a system. It works in conjunction with the directory node. Think of the directory as the repository that holds:

- Data about security, such as user IDs and associated passwords, or digital certificates (used to authenticate a user)
- Lists of services that a user is authorized to perform (authorization or access control)

Think of the security node as holding the set of components that define the decisions to be made. The node may perform the actual security processing, for example, verify certificates. The authentication in most current designs validates the access to the Web Application Server, but it can also authenticate the access to the Database Server. The security domain describes the components needed to implement the technical dimension of security and how these components interact to implement the technical aspects of a security policy. The components that implement security are distributed throughout the network. It is unlikely there is a node in the system that does not include some components implementing some aspect of security. The security node represents centralized services that support security on other nodes.

Self-Service Business pattern Captures the essence of direct interactions between interested parties and a business.

Sell-Side Hub Composite pattern An e-Marketplace owned by the seller where the seller uses it as a vehicle to sell goods and services to prospective buyers across the Web.

Server A process offering some service and either driven by RPC, a session, or a queue. In the context of a X-Window System, it is a graphics application that controls screen presentation according to messages from a client application.

Shared database A single database that is shared across multiple transactional Web servers.

Shared file server node The timely synchronization of several Web servers is achieved by using a shared file system as the content storage and capitalizing on the replication capability of this technology.

Single Sign-On Application pattern Provides a structure for integrating several applications under a portal that provides single sign-on capability and role-based access to certain information and applications.

Stand-Alone Single-Channel Application pattern Provides a structure for applications that have no current need for integration with other systems and need only focus on one delivery channel.

Store and Retrieve Application pattern Allows users to collaborate with others in the network interactively whether the communicating partners are online simultaneously or not.

SMTP Simple Mail Transfer Protocol.

SOAP Simple Object Access Protocol provides a way for applications to communicate with each other over the internet, independent of platform. Unlike DCOM's IIOP, SOAP piggybacks a DOM onto HTTP (port 80) in order to penetrate server firewalls, which are usually configured to accept port 80 and port 21 (FTP) requests. SOAP relies on XML to define the format of the information and then adds the necessary HTTP headers to send it. SOAP was developed by Microsoft, DevelopMentor, and Userland Software and has been proposed to the Internet Engineering Task Force (IETF) as a standard. For more information go to: `http://www.develop.com/soap/soapfaq.htm`.

Solaris Solaris refers to the operating system that controls many of the machines provided by Sun Microsystems Corp.

Solution Overview Diagram A graphic representation of the Actors, Business Functions, and Interactions between them identified from a Business Requirements Specification.

Source data Source data is the data store from which the extract process reads its input data. It is a generic term representing either operational data, BDW, data mart, ODS, or information catalog when they are being read from.

SQL A standard interactive and programming language for getting information from and updating a database. Although SQL is both an ANSI and ISO standard, many database products support SQL with proprietary extensions to the standard language. Queries take the form of a command language that lets the user select, insert, update, and discover the location of data and so forth. There is also a programming interface.

SSL Secure Sockets Layer (SSL) refers to encryption technology, which is commonly used between the browser on a user's PC and a Web server node. It is used to protect the data in messages from being viewed in an un-authorized fashion while travelling over a TCP/IP network. It can also continue to protect the messages as they flow over internal TCP/IP networks between nodes after passing the Web server.

Supply chain management (SCM) A supply chain is the way an enterprise ties together internally and externally the people, processes, and related information associated with its flow of goods. Supply chain management is a strategic approach that unites all steps in the cycle, from initial product design and the procurement of raw materials through production, shipping, distribution, and warehousing, until a finished product is delivered to a customer.

Synchronous In the context of system modes of interaction, the interaction is consecutive or serial with involved components being active at the same time. That is, components issue a request and wait for something to happen.

System The computer based parts of one or more applications, including data and infrastructure.

Systems integrator The company or person(s) responsible for the integration of purchased systems (often including hardware and software) into an existing IT environment.

Systems management architecture Defines the approach for monitoring, administration, and management of the applications and the infrastructure.

Systems management node This node is a logical representation of the functions required to manage all the nodes and components in the system, including the management of problems, changes, performance, configuration of assets, and others. There are usually two aspects to systems management:

- A managing aspect (with server components running on one or more systems management servers)
- A managed aspect (with systems management client components running on every node in the system)

There should be client components running on every node, which is able to accept and install changes sent from a change management server. The changes may be either pulled from the client on demand, or pushed from the server with centralized control. On critical nodes, there may also be problem management components that send a "heartbeat" back to a central monitoring site. If a heartbeat is missed, the managing site raises an alert.

Target data The data store to which the load process writes its output data. It is a generic term representing either BDW, data mart, ODS, or the information catalog when they are being written to.

TCP Transmission Control Protocol.

TCP/IP Transmission Control Protocol / Internet Protocol.

Technology architecture Defines the technology services needed to support the application portfolio of the business. It also documents the software, hardware, and network product standards.

Tier A logical layer within a design that allows the subdivision of the application into major functional collections. Tiers can be further subdivided into functional components.

Tier-to-tier connection and label A connection between two tiers indicates that a programmatic interaction takes place between the two tiers. It is labeled synchronous or asynchronous to indicate whether the programmatic interaction between the adjacent tiers takes place synchronously or asynchronously.

Thin Client Transactional pattern The Enterprise Solution Structure (ESS) technical architecture that addresses the need to do enterprise-scale administrative business, as opposed to solutions requiring real-time control of equipment, for example, Customer sales and service, and Order processing, Claims processing, loan origination, and so on. Its purpose is to support the business need of doing enterprise-scale commerce (as contrasted with business intelligence or collaboration) over the Web or through network-connected workstations. The essence of this pattern is the need to use highly secure, highly scalable transaction processing through this new channel.

Transaction An exchange, often between two computers, involving a request for information or the execution of a function and the response to the request. Also see **Unit of Work**.

Transactional workflow manager Primarily concerned with driving a sequence of system-level transactions that constitute some higher level (business) process or task. A transaction workflow manager focuses on process logic, including any error related corrective or compensatory action, but may also include functionality to provide direct transaction coordination (such as two-phase commit) across attached cooperating systems.

Transcoding Bridges data across multiple formats, markup languages, and devices. In addition, it adapts, reformats, and filters content to make it suitable for pervasive devices. Products such as WebSphere Transcoding Publisher implement this technology. More information can be found at `http://www.ibm.com/software/Webservers/transcoding/`.

Transform Includes any processes used in data acquisition or data propagation that convert data from one form to another by the application of rules, lookup tables, or combination with other data.

Transform + (load) A combination in a single process of any transform function required and load function.

Two-phase commit A protocol that allows a set of autonomous processes or agents to eventually all commit or all abort a particular transaction or set of transactions. Most transaction managers offer to act as the coordinator in a two-phase commit protocol. Variants of the protocol are found in LU6.2 and OSI-TP.

UML The Unified Modeling Language (UML) is the industry-standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. The UML notation is an evolution from older OO notations, now endorsed as a standard for OO modelling development by the Object Management Group (OMG).

Unit of Work A transaction is a recoverable unit of work, or a group of SQL statements that can be treated as one atomic operation. This means that all the operations within the group are guaranteed to be completed (committed) or undone (rolled back) as though they were a single operation. A transaction can also be referred to as a Unit of Work or a Logical Unit of Work. When the transaction spans multiple connections, it is referred to as a Distributed Unit of Work.

A single unit of work refers to the (one or more) SQL COMMITs for creating, updating, or deleting, for a single business transaction that must be applied to multiple tables, distributed relational databases, and so on. and that are done in conformance with the ACID properties of transaction semantic integrity, that is, atomicity, consistency, isolation, and durability.

UNIX® The multi-user, multitasking operating system developed at Bell Labs in the early 1970s.

User experience .Identifies the different user groups and their user experience needs. Also defines enterprise-wide presentation standards.

User facing An application system that directly interfaces with the user.

User-to-Business pattern Now known as the Self-Service Business pattern, it is the general case of users (internal to the enterprise or external) interacting with enterprise transactions and data. It is relevant to those enterprises that deal with goods and services not normally listed in and sold from a catalog. It covers all user-to-business interactions not covered by the User-to-Online Buying pattern. This Business pattern also covers the more complex case where there is a need to access back-end applications and data.

User-to-Data pattern Now known as the Information Aggregation Business pattern, it encompasses the provision of Business Intelligence (BI) capabilities to an organization.
The user is someone connected to the data through one of four paths:

- Internet: The user is external to the company, or an agent of the company.
- Intranet: A staff (internal) user.
- Extranet or privileged internet: an associate of the client's organization who acts as a business agent on the company's behalf.
- Fat client-connected as in a client-server system: Applies to internal users only.

The data can be held in a:

- Web-content store: Holding Web pages and cached information. The data may include copies of operational detailed records, such as consolidated account information for a customer reference. The data is read-only; if an update of the data is required, the pattern is User-Business, not User-Data.
- Data mart: The data may be read-write with local scope-of-effect only.
- Data warehouse: The data is read-only for applications.
- Tool-specific store (proprietary): Some tools, such as Essbase, require specialized data stores for efficiency.

User-to-Online Buying pattern A special case (now part of Self-Service Business pattern) where products are sold through a catalog using a shopping cart, a wallet, and so forth. This business pattern can also include links to back-end systems to allow for inventory updates and credit checking. Examples of the User-to-Online Buying pattern:

- Consumers purchasing goods online.
- Buyers purchasing goods online from a supplier.

User-to-User pattern Now known as the Collaboration Business pattern, it describes users collaborating with one another by e-mail, shared documents, and so on. For example, collaborating across teams on document development.

Value Added Network A VAN is a networking service that leases communication lines to subscribers and adds extra services or capability such as security, error detection, guaranteed message delivery, and a message buffer.

VPN endpoint A Virtual Private Network (VPN) is an extension of an enterprise's private intranet across the internet or other public network. It creates a secure private "tunnel" through the internet to the other partner. You can create configurations that access the VPN from behind the domain firewall.

WAN Wide Area Network.

WAP (Wireless Application Protocol) A standard for providing cellular phones, pagers, and other handheld devices with secure access to e-mail and text-based Web pages. Introduced in 1997 by Phone.com (now Openwave Systems), Ericsson, Motorola, and Nokia.

WAP provides a complete environment for wireless applications that includes a wireless counterpart of TCP/IP and a framework for telephony integration such as call control and phone book access.

WAP features the Wireless Markup Language (WML), which was derived from Phone.com's HDML and is a streamlined version of HTML for small screen displays. It also uses WMLScript, a compact JavaScript-like language that runs in limited memory.

WAP also supports handheld input methods such as a keypad and voice recognition. Independent of the air interface, WAP runs over all the major wireless networks in place now and likely into the future. It is also device independent, requiring only a minimum functionality in the unit so that it can be used with a myriad of phones and handheld devices.

Web application server node A Web application server node is an application server that includes an HTTP server (also known as a Web server) and is typically designed for access by HTTP clients and to host both presentation and business logic.

The Web application server node is a functional extension of the informational (publishing-based) Web server. It provides the technology platform and contains the components to support access to both public and user specific information by users employing Web browser technology. For the latter, the node provides robust services to allow users to communicate with shared applications and databases. In this way, it acts as an interface to business functions, such as banking, lending, and HR systems.

Web mart A Web mart is a data mart designed for user access through a Web browser and optional plug-ins. Data in the Web mart is stored in a standard Internet format such as HTML, and resides on a server in the DMZ.

Web server A process that supports the use and management of Web pages and other internet-related material.

WebSphere Application Server Provides an e-business application runtime environment based on open standards. More information can be found at `http://www.ibm.com/software/webservers/appserv/`.

WebSphere Business Integrator Enables you to seamlessly span the gap between your own enterprise computing systems and those of your customers, suppliers, Business Partners, and marketplaces simultaneously. More information can be found at `http://www-ibm.com/software/webservers/btobintegrator/`.

WebSphere Everyplace® Suite A comprehensive, integrated software platform for extending the reach of e-business applications, enterprise data, and internet content into the realm of pervasive computing. More information can be found at `http://www.ibm.com/pvc/products/wes/`.

WebSphere Partner Agreement Manager Enables multiple partners to participate in shared business processes that can be partially or fully integrated depending, on business needs. More information can be found at `http://www.ibm.com/software/webservers/pam/`.

WML Wireless Markup Language is an XML language used to specify content and user interface for WAP devices; the WAP forum provides a DTD for WML. WML is supported by almost every mobile phone browser around the world. WML pages are requested and served in the same way as HDML pages. For Web servers to serve WML pages, they must contain the text/vnd.wap.wml mime type.

WebSphere Translation Server A machine-translation offering that can help companies remove language as a barrier to global communication and e-commerce. More information can be found at `http://www.ibm.com/software/speech/enterprise/ep_8.html`.

Workflow An emerging concept that centers on the problem of tracking and managing work in a team environment. Workflow is a manual or automated approach for routing this work from one individual to another.

Workflow manager A workflow manager provides model-driven business process automation and tracking, involving systems or systems and people. These processes can span multiple applications and organizational boundaries. A workflow manager maintains state and tracks sequencing through the business process to guide what the application or user does with a message as it arrives.

Workflow server node The function of the workflow server node is to manage the flow of operations for users and applications within the e-Marketplace. The scope of workflow management is categorized into the groups macro and micro:

- Macro: At the macro level, the workflow server manages functions at a broad level, such as governing a complete business process.
- Micro: At the micro level, the workflow server manages lower-level tasks, such as the flow of user interaction associated with an online product purchase.

XML See Extensible Markup Language

XSL See Extensible Stylesheet language

Zone Represents an area for which a common set of non-functional requirements can be defined.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this IBM Redbooks publication.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 226.Note that some of the documents referenced here may be available in softcopy only.

- *Access Integration Pattern using IBM WebSphere Portal Server*, SG24-6267
- *Applying the Patterns for e-business to Domino and WebSphere Scenarios*, SG24-6255
- *Mobile Applications with IBM WebSphere Everyplace Access Design and Development*, SG24-6259
- *Patterns: Connecting Self-Service Applications to the Enterprise*, SG24-6572
- *Patterns for the Edge of Network, SG24-6822*
- *Self-Service Applications using IBM WebSphere V4.0 and IBM MQSeries Integrator, SG24-6160*
- *Self-Service Patterns using WebSphere Application Server V4.0*, SG24-6175

Other publications

These publications are also relevant as further information sources:

- Adams, et al, *Patterns for e-business: A Strategy for Reuse*, Mc Press, 2001, ISBN 1931182027
- Gamma, et al, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994, ISBN 9780201633610

Online resources

These Web sites and URLs are also relevant as further information sources:

- The Federal CIO Council Web site:

  `http://www.cio.gov/Douments/fedarch1.pdf`

- Patterns for e-business Web site:

  `http://www.ibm.com/developerWorks/patterns/`

- Patterns for e-business Product mappings Web site:

  `http://www-106.ibm.com/developerworks/patterns/select-pattern.html`

- Patterns for e-business, WebSphere and Domino Integration Web site:

  `http://www-106.ibm.com/developerworks/patterns/ws-domino/index.html`

- Patterns for e-business, WebSphere and SAP Integration Web site:

  `http://www-106.ibm.com/developerworks/patterns/sap/select-runtime-topology.htm`

- Unified Modeling Language (UML) and Use Case Modeling Web site:

  `http://www.omg.org`

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy IBM Redbooks publications or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM Redbooks

Academic Edition: Applying Pattern Approaches Patterns for e-business Series

(0.2"spine)
0.17"<->0.473"
90<->249 pages

IBM®

Academic Edition:
Applying Pattern Approaches Patterns for e-business Series

Redbooks

Applying Patterns for e-business effectively

Integrating pattern approaches within a methodology

Validating technical scenarios with case studies

As companies compete in the e-business marketplace, they find that they must re-evaluate their business processes and applications so that their technology is not limited by time, space, organizational boundaries, or territorial borders. They must consider the time it takes to implement the solution as well as the resources (people, money, and time) they have at their disposal to successfully execute the solution. These challenges, coupled with the integration issues of existing systems and the pressure to deliver consistent high-quality service, present a significant undertaking when developing an e-business solution. IBM® Patterns for e-business provide a structured way of decomposing simple or complex requirements into a layered set of reusable architectures and designs.

This IBM Redbooks publication's focus is on documenting the currently observed usages of the Patterns for e-business. This IBM Redbooks publication is part of the Patterns for e-business series. We introduce three identified pattern approaches and examine pattern scenarios for each of them. The three pattern approaches are:

Approach 1: Building solutions on the basis of a Single Business pattern

Approach 2: Building solutions on the basis of Multiple Business patterns

Approach 3: Applying Patterns with existing Custom designs

We discuss the key features and identify guidelines for each of these approaches. We document the alignment of these approaches within a context of a proven methodology, the IBM Global Services Method.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7466-00 ISBN 0738486612