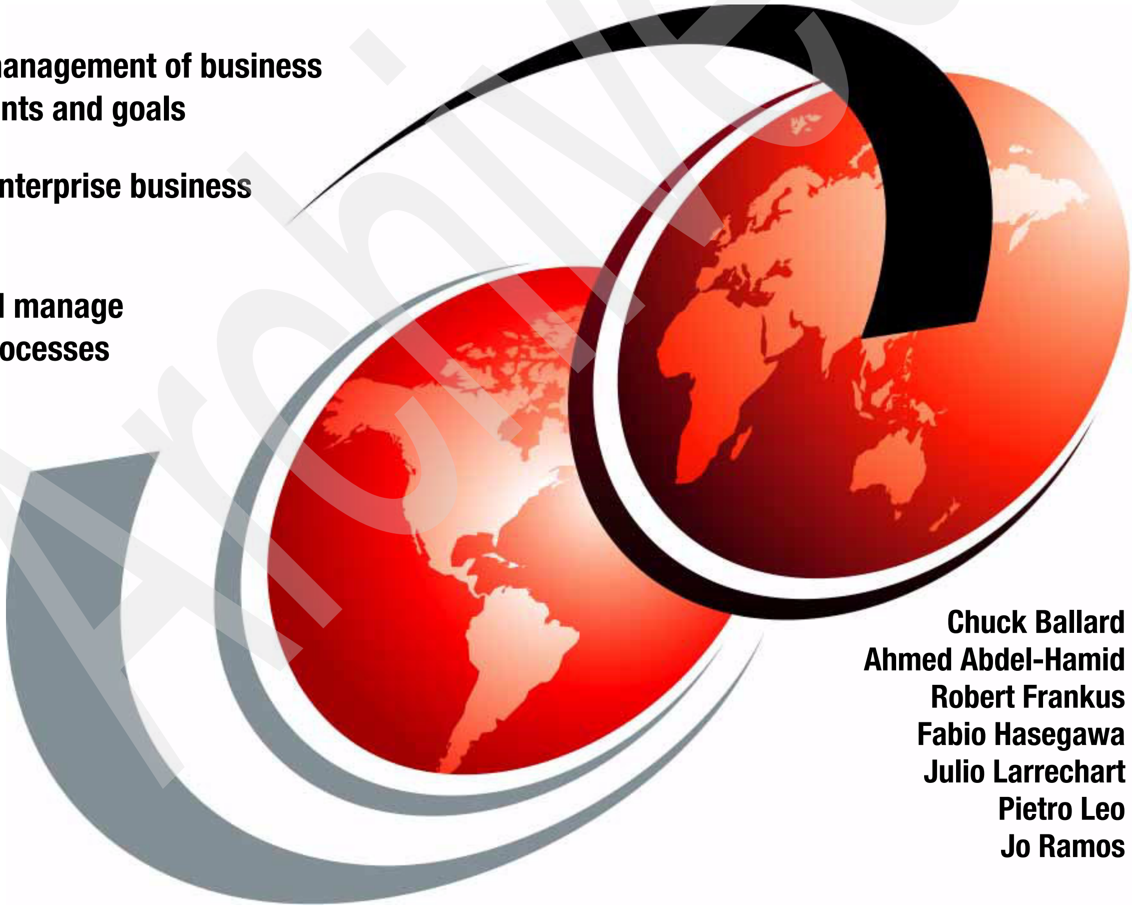
DB2® Information Management Software

IBM

Improving Business Performance Insight . . .
with Business Intelligence and Business Process Management



Proactive management of business measurements and goals

Real-time enterprise business intelligence

Monitor and manage business processes

Chuck Ballard
Ahmed Abdel-Hamid
Robert Frankus
Fabio Hasegawa
Julio Larrechart
Pietro Leo
Jo Ramos

Redbooks

ibm.com/redbooks

International Technical Support Organization

Improving Business Performance Insight . . . With Business Intelligence and Business Process Management

August 2006

SG24-7210-00

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

First Edition (August 2006)

This edition applies to DB2 Data Warehouse Edition V9, DB2 ESE Version 8.2, DB2 Alphablox Version 8.3, WebSphere Information Integrator V8.3, WebSphere Business Modeler V6.0.1, WebSphere Business Monitor V6.0.1, WebSphere Portal Server V5.1, WebSphere Application Server V6, WebSphere Process Server V6, WebSphere Message Broker V6, AIX Version 5.2, Windows 2000, and Red Hat Linux Enterprise Edition Version 3.

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 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 products and/or the programs 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 illustrates 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. 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 IBM's application programming interfaces.

Trademarks

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

- eserver®
- Redbooks (logo) ™
- iSeries™
- pSeries®
- xSeries®
- z/OS®
- z/VM®
- zSeries®
- z9™
- AIX®
- ClearCase®
- Cube Views™
- CICS®
- Database 2™
- DataStage®
- Distributed Relational Database Architecture™
- Domino®
- DB2 Connect™
- DB2 Universal Database™
- DB2®
- DRDA®
- HiperSockets™
- Informix®
- Intelligent Miner™
- IBM®
- IMS™
- Lotus Notes®
- Lotus®
- MetaStage®
- MQSeries®
- Notes®
- OS/400®
- PR/SM™
- QuickPlace®
- Rational®
- Redbooks™
- Sametime®
- System z9™
- Tivoli®
- TotalStorage®
- Virtualization Engine™
- WebSphere®
- Workplace™
- Workplace Web Content Management™
- XDE™

The following terms are trademarks of other companies:

Enterprise JavaBeans, EJB, Java, Java Naming and Directory Interface, JavaBeans, JavaScript, JavaServer, JavaServer Pages, JDBC, JDK, JSP, JVM, J2EE, Solaris, Streamline, Sun, Sun Microsystems, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Excel, Microsoft, Visio, Windows NT, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

This IBM Redbook is primarily intended for use by IBM Clients and IBM Business Partners. In it, we describe and demonstrate IBM Business Innovation and Optimization (BIO). However, we have a particular focus on performance insight, which is one of the on-ramps of BIO. Performance insight is, simplistically put, the integration of business process management and business intelligence. It is, of course, business performance insight. We have shortened the term many times throughout this redbook to performance insight. But, those terms are synonymous.

The objective of performance insight is to enable the design, development, and monitoring of the business processes and to make the data from those processes available for business intelligence purposes in near real-time. This, in turn, can better enable management to proactively manage the business processes through improved insight into the business performance. Doing that means avoiding problems and issues, rather than reactively trying minimize their impact. Then companies can really focus on meeting their performance objectives and business goals.

Business process management is an initiative for the effective use of people, processes, assets, and technology to proactively achieve business measurements and goals. It enables strategic alignment of business and technology, resulting in real-time access to data and continuous process and data flow, for proactive business management. By pursuing performance insight, organizations can better understand how people, processes, assets, and technology can be used together across the enterprise.

The integration with business intelligence (BI) is key to success. In this redbook, we discuss the techniques, architectures, and processes for implementing and integrating BI to achieve business performance insight. Among the specific techniques and technologies used are key performance indicators, alerts, management dashboards, analytic applications, application integration, process modeling and monitoring, and real-time business intelligence. The products featured are DB2 UDB, DB2 Alphablox, WebSphere Information Integrator, WebSphere Portal, WebSphere MQ, WebSphere Message Broker, WebSphere Process Server, WebSphere Business Monitor, and WebSphere Business Modeler.

With BI, we want to demonstrate how this integration can better enable a more proactive, in addition to the more typical reactive, form of business intelligence. And that is what enables fast action to be taken to resolve issues and actually drive the attainment of measurements and goals rather than simply passively monitoring their status.

For example, we demonstrate the capability to actively monitor the business processes and integrate that status data with the operational activity data in the data warehouse. The combination of these two sources of data provides an enterprise-wide view of the business for decision making. With this information, we can begin to manage and optimize business performance. This is significant value for the enterprise, business management, and business shareholders.

Business process management has been developed to monitor and manage business processes with the objective of enabling improved business performance. It has three primary categories of capability. These capabilities are discussed in more detail throughout this redbook:

- Information Management: including operational reporting, data federation, data warehousing, and business intelligence
- Process Management: including business processes, key performance indicators (KPI), alerts, process status, operational activities, and real-time process monitoring and data acquisition
- Business Service Management: including systems monitoring and optimization of IT operations to meet the business measurements

In general, there can be several points of integration such as visual, data, and functional. In this redbook we focus on visual and data. Thus, the results of these capabilities are brought together at the business portal, our point of integration for management and decision makers.

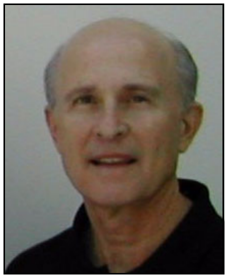
As businesses move forward in the evolution to real-time business intelligence, there is a need to optimize the operational business activities. For example, they must be modified to support real-time activity reporting and continuous flow of the data to the enterprise information repository - the DB2 data warehouse. One major impact of this evolution is enhanced decision making, and proactive avoidance of problems and issues in addition to more typical reactive measures to minimize the impact of those problems.

Products, such as WebSphere Process Integration Suite and DB2 UDB, play a key role in business process management and were used in the development of this redbook. We have included example system architectures, product installation and configuration examples and guidelines, examples of the use of key performance indicators (KPI), and management dashboards to enable improved business process management. We believe this information and our examples will be of great benefit as you continue to improve the proactive management of your business performance.

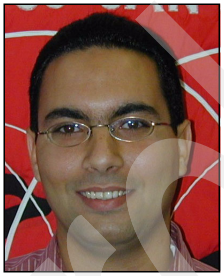
The team that wrote this redbook

This redbook was produced by a team of business and technical specialists from around the world working at the International Technical Support Organization, San Jose Center.

The team members are depicted below, along with a short biographical sketch of each:



Chuck Ballard is a Project Manager at the International Technical Support organization, in San Jose, California. He has over 35 years experience, holding positions in the areas of Product Engineering, Sales, Marketing, Technical Support, and Management. His expertise is in the areas of database, data management, data warehousing, business intelligence, and process re-engineering. He has written extensively on these subjects, taught classes, and presented at conferences and seminars worldwide. Chuck has both a Bachelors degree and a Masters degree in Industrial Engineering from Purdue University.



Ahmed Abdel-Hamid is a Staff Software Engineer in the Cairo Technology Development Center, IBM Egypt. As part of WebSphere Business Monitor development team, his areas of expertise cover dashboards and WebSphere Portal development, DB2 Alphablox Analytics, and the Process Integration Suite. Before joining IBM, his experience was in the development of computer telephony and telecommunication applications. Ahmed holds a Bachelors degree in Computer Engineering and is pursuing his Masters degree in the field of Computer Graphics in Faculty of Engineering, Cairo University.

Robert Frankus is a Senior IT Specialist for the IBM Business Intelligence Best Practices Team, San Francisco, CA. His areas of expertise are in analytical applications, application integration, and information management integration. Over the last seven years, he has architected and developed a number of custom analytical applications, dashboards, and corporate performance management solutions for Fortune 500 clients, in industries such as retail, high tech and financial services. He teaches extensively on dimensional modeling and building business intelligence applications. He holds a Masters of Management Information Systems from the University of Cologne, Germany and was a Visiting Fellow at the Massachusetts Institute of Technology Sloan School of Management, Boston, Massachusetts.

Fabio Hasegawa is a Senior IT Specialist leading the DBA Distributed Services Center, IBM Application Management Services, Brazil. He has been mastering several IBM products, such as DB2, WebSphere Application Server, WebSphere Message Broker, and WebSphere Information Integration. He has been working for 10 years in IT. During this time, he has worked on several projects helping Brazilian clients in various segments (telecommunciations, financial, government, and banking). Fabio's areas of specialty are infrastructure sizing, performance management, and designing and improving high availability solutions focused on information management services.

Julio Larrechart is a Business Analyst at IBM Business Consulting Services, Application Services in Uruguay. He has a Information System Engineering degree from Engineering University in Argentina and post-degree studies in Computing Engineering from Engineering University in Uruguay. His areas of expertise include Business Process Management, Requirement Management, and Social Security industry. He has experience in Business Intelligence with the TACS (Tax Audit and Compliance System) tool.

Pietro Leo is a Senior IT Architect at IBM Business Consulting Services in Bari, Italy. He is a permanent member of the IBM Italy Technical Expert Council affiliated with the IBM Academy of Technology. His areas of expertise include Data, Information and Application Integration, Unstructured Information Management, Mining and Semantic/Conceptual Indexing and Search, Bioinformatics, Healthcare, and Wireless Solutions. Pietro holds a higher artistic degree in Oboe from the Music Conservatoir of Lecce (Italy); a Computing Science degree from the University of Bari (Italy); an Advanced Computing Science degree from University of Udine (Italy); a Master of Science by Research degree in Applied Artificial Intelligence from University of Aberdeen (UK), and a Masters degree in Public Funding Management for Business from Tax Consulting Firm (Italy). Pietro has been an invited speaker at Industrial and Scientific Conferences as well as a member of Scientific Committees of Conferences. He is the author or co-author of more than 40 scientific/technical publications for Journals, has presented during National and International Conferences, and is the co-author of an IBM Redbook on XML.

Jo Ramos is a Business Intelligence (BI) specialist for the BI Best Practices team in Dallas, Texas. He has eight years of experience with customer BI projects, providing support in consulting, architecture design, data modeling, and implementation of data warehouses and analytical solutions. Jo has worked with the majority of the IBM and Information Management products portfolio, specializing in analytical solutions. He holds a BS degree in Economics from the FURB University in Santa Catarina, Brazil.

Other Contributors:

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

From IBM Locations Worldwide

- Catherine J. Burton, Value Creation Leader, IBM Center for Business Optimization, Raleigh, NC
- Brenda Dietrich, IBM Research Director, Mathematical Sciences, IBM T J Watson Research Center, Yorktown Heights, NY
- David Enyeart, Business Innovation and Optimization Architecture, Durham, NC
- Jean-Poul Jacob, IBM Emeritus, Chair of IBM Almaden's University Relations Committee and as IBM's Campus Relationship Manager for UC-Berkeley, Berkeley, CA
- Guy Lohman, IBM Research Manager, Advanced Optimization and Autonomic Computing Almaden, San Jose, CA
- Melissa Montoya, DB2 Information Management Skills Segment Manager, Menlo Park, CA
- Kramer Reeves, Product Marketing, Business Innovation and Optimization, Durham, NC
- Billy Rowe, Business Innovation and Optimization Architecture, Durham, NC
- Guenter Sauter, Information Integration Solutions Architect, Somers, NY
- Paula Wiles Sigmon, Product Marketing Director, Information Integration Solutions, Somers, NY
- James Spohrer, IBM Research Director, Almaden Services Research, Almaden, San Jose, CA
- Robert Spory, Solutions Sales Executive, Business Innovation and Optimization, Hazelwood, MO

- Louis Thomason, STSM, Information Integration, Somers, NY
- Ueli Wahli, Project Leader, ITSO, San Jose, CA

From the International Technical Support Organization, San Jose Center

- Mary Comianos, Operations and Communications
- Deanna Polm, Residency Administration
- Emma Jacobs, Graphics
- Leslie Parham, Editor

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review redbook form found at:

  `ibm.com/redbooks`

- Send your comments in an e-mail 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

IBM is a leader in information technology, providing the hardware and software to enable businesses to gain a competitive advantage. With research and years of experience, IBM has always provided robust frameworks and architectures for business initiatives, that are well thought out, complete, and that stand the test of time. Well, here we go again.

In the fast-moving marketplace of today, businesses must remain flexible and agile to maintain a competitive advantage - and perhaps even to remain viable. Shorter business measurement cycles, and stakeholders that demand business goals and performance measurements are met, are putting increasing pressure on business management.

Managing and optimizing business performance is a critical requirement for maximizing business profitability, returning shareholder value, and gaining a competitive advantage. To do this requires monitoring and tracking capabilities that can generate current, complete, and accurate information upon which management can immediately act. Businesses are looking for help in developing the capability to meet these new demands. And it is here.

1.1 Business Innovation and Optimization

IBM has again stepped up to the challenge with an initiative that we call business innovation and optimization (BIO). It is an approach from IBM that enables organizations to understand the state of their business and gives them the tools to take action in response to changing business conditions.

This is not a market category, but rather represents a collection of software technology capabilities, best practices, and industry expertise to address needs and functionality identified in several market categories including business performance management, enterprise performance management, corporate performance management, business intelligence, business services management, business process management, and business activity monitoring.

Highly effective organizations employ common and simplified processes, as well as data and information standards. They minimize the number of disparate technologies, platforms, and systems used, improve the integration of those systems, and focus on the delivery of valuable business insight. These types of organizations are more likely to have standard policies, common and simplified processes, functional best practices and the appropriate supporting technology.

As innovative organizations, they integrate role-based metrics and exception reporting while using those metrics consistently throughout their enterprise. They define quantifiable relationships between business drivers, scorecards, and dashboard metrics to proactively monitor and effectively manage their performance against the business goals and measurements. They are doing this by moving from a role of static reporting and data stewardship to a more predictive role of providing dynamic business insight to decision makers.

BIO combines advanced software technologies with industry-specific expertise and best practices. It is an action-oriented approach to help organizations:

- Gain real-time insight into the state of their business.
- Take action to become more predictive, mitigate problems, and achieve faster growth.

It does this by providing effective business strategy execution and business operations management. It helps organizations become more aligned, accountable, and action-oriented in order to achieve their business goals.

Effectively using people, processes, assets, and technology to achieve strategic alignment, continuous improvement, and innovation is the objective. By implementing a business innovation and optimization strategy, organizations can better understand and respond to how people, processes, assets, and technology act together across the enterprise.

1.1.1 BIO on-ramps

The BIO approach is both deep and broad. To make getting started with BIO manageable, there is a set of what we call *On-Ramps*. On-Ramps represent a logical segmentation of common business goals that were identified through analyst feedback, market research, and customer engagements that we can address today. Those are:

- Process Innovation: to deliver continuous innovation and improvement
- Performance Insight (or Business Performance Insight): to enable more effective decision making
- Operational Management: to manage business operations effectively
- Business Alignment: to align business objectives

The BIO on-ramps are depicted in Figure 1-1.

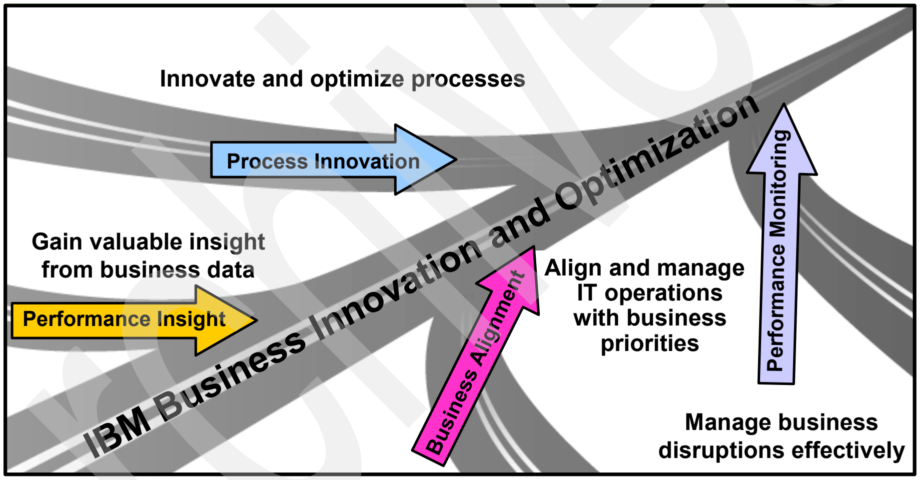


Figure 1-1 BIO on-ramps

The scope of what can be accomplished with the BIO approach is too broad for this redbook, so we are going to focus on the core products to develop a solution for the *Performance Insight* on-ramp.

1.1.2 Performance Insight on-ramp

Performance insight is about taking a holistic approach for managing business performance. Businesses align strategic and operational objectives, and business activities, to fully manage performance through more informed and proactive decision making.

This is achieved by optimizing decision making with real-time contextual insight to take action faster. One way to enable this is through the use of dashboards to provide the right information to the right people at the right time. The real-time updates can be used to reduce the lag time between execution and understanding and enable faster action to be taken.

This information can also be used to analyze process metrics and update the process model with observed results to run future simulations. The holistic approach is enabled through the integration and use of business process management and business intelligence. We start with business intelligence to understand past actions and performance and to provide direction for future actions.

To enable this capability requires changes, not only to the software systems, but to the business processes themselves. Unless there is adoption of common processes and standards for the enterprise, the primary option is management by human intuition rather than by facts. This means relying on the manual effort of a limited number of analysts to maintain information integrity.

Such an environment brings with it all the shortcomings of a manual solution. Those include difficulty of knowledge transferability, inconsistency, lack of efficiency, and a continual need to verify the data relevancy before using it to make business decisions. And such a procedure is indeed difficult to optimize and standardize. It would be much better to embed this into a controlled set of processes. But these critical business processes must be implemented on a firm foundation.

The operational processes and activities that support the enterprise business strategy must be defined for accuracy, consistency, efficiency, and effectiveness. But, they also need to be flexible and adaptable to change. To support business process management, business activities must include facilities to monitor and track the process execution across the business value chain. These activities must gather process status and operational activity execution data and deliver relevant, timely, and actionable information to the business decision makers. This information provides the required business intelligence for proactively managing business performance.

There is another redbook on a related topic, and more specifically on business performance management, that you might want to reference. It is called *Business Performance Management . . . Meets Business Intelligence*, SG24-6340. For details about obtaining that redbook, refer to “Related publications” on page 427.

1.2 Overview and objectives

As stated, we want to describe and demonstrate IBM Business Innovation and Optimization (BIO). But, in particular, we will focus on performance insight. As you might imagine, it also involves a number of other related initiatives. Some examples are information integration, evolving to a real-time environment, automation of enterprise-wide consistent, but dynamic, business processes, integration of IT and the business processes, proactive (push) delivery of information to decision makers, and ongoing monitoring of the business processes to enable proactive alerting based on key business performance metrics.

To help us satisfy that objective, we describe and demonstrate the integration of the business processes (or business process management) with business intelligence. It is this combination that enables us to deliver performance insight. We discuss the techniques, architectures, and processes for implementing and integrating BI and business process management to achieve performance optimization. Among the specific techniques and technologies used are key performance indicators, alerts, management dashboards, analytic applications, application integration, process modeling and monitoring, and real-time business intelligence. The products featured are DB2 UDB, DB2 Alphablox, WebSphere DataStage, WebSphere Information Integrator, WebSphere Portal, WebSphere MQ, WebSphere Message Broker, WebSphere Process Server, WebSphere Business Monitor, and WebSphere Business Modeler.

Wow, that sounds like quite a number of initiatives and products to cover. And, it is. But you will find the journey, and the information you gather along the way, significantly valuable. So, let us get started.

1.3 Contents abstract

This section includes a brief description of the topics presented in the redbook, and how they are organized. The information presented ranges from high-level product and solution overviews to detailed technical discussions. We encourage you to read the entire book. However, depending on your interest, level of detail, and job responsibility, you might want to be selective and focus on those topics of primary interest. We have organized this redbook to accommodate readers with a range of interests.

Summary by chapter

Here we provide a brief overview of the contents of each chapter of the redbook to help you in your reading selections:

- **Chapter 1** introduces the IBM approach for understanding the overall state of the business and enabling proactive actions for optimizing business performance. This is called *business innovation and optimization* (BIO). It is a robust enterprise approach consisting of four primary *on-ramps*. These on-ramps represent the logical segmentation of common business goals. This redbook has a focus on the specific on-ramp called *Performance Insight*.
- **Chapter 2** provides a more detailed description of IBM Business Innovation and Optimization (BIO). We first provide an overview and then introduce the primary strategy and components. A description of the BIO approach is provided to help with a fast start. Optimization is a key topic here because the objective is to meet the performance goals and measurements of the business. The IBM software products are mapped to the BIO functions to enable optimization of the processes as we gain insight into them.
- **Chapter 3** continues with an introduction of the Performance Insight on-ramp. From a simplified perspective, performance insight is the integration of business process management and business intelligence. We describe the different elements of performance insight, and how it enables improved decision making. It is this faster and improved decision making, and new business insight, that enables organizations to be more proactive and better able to meet their business performance measurements and enterprise business goals.
- **Chapter 4** provides a discussion of how the business of the organization can be perceived through processes. We can then begin to understand what a process is and the elements which comprise it. The information helps provide a conceptual framework for how organizations can describe and define their businesses to be focused in processes. Along with that, we present some of the advantages and define solution content for business process management.
- **Chapter 5** focuses on the current stages of development of data warehousing and business intelligence solutions in organizations and the extent to which those solutions are being used to support strategic, tactical, and operational decisions.
- **Chapter 6** presents and describes the product components used in the redbook case study outlined in Chapter 7, “Performance insight case study overview” on page 247. We have included an expanded and detailed set of component descriptions here for your benefit. There are a significant number of components, and it is important that their framework and integration needs are well understood. This is a relatively new initiative, and requirements are frequently updated.

- **Chapter 7** provides the description of the redbook case study to demonstrate performance insight as it could be used in practice. The redbook case study was developed using a number of the IBM products and technologies that are positioned to support BIO. We highlight and use some of those capabilities, but certainly not all of them. That would be well beyond the scope of this redbook.
- **Chapter 8** describes and demonstrates the implementation of the performance insight case study solution. As a brief summary, we described the processes, defined them with WebSphere Business Modeler, implemented them, executed them in the WebSphere Process Server, monitored their execution with WebSphere Business Monitor, and demonstrated how we gained performance insight to identify and resolve business problems. It is a closed-loop process that results in resolving the problems and modifying the processes to minimize the risk of recurrence.

With that overview of the contents of the redbook, it is now time to make your reading selections and get started.

2

Business innovation and performance optimization

The word *innovation* has recently become a key word in a number of business strategies and initiatives. It seems everyone is now using it. At the time of the writing of this book, Spring 2006, we just did a Google search on the Web for innovation, and it returned more than 335,000,000 hits!

In general, this word seems to have one primary benefit for businesses - and that is survival. By that we mean that they must be able to respond quickly, intelligently, and give their customers what they want, when they want it - if they want to be competitive, or to simply survive. In this information age, innovation is becoming a requirement to succeed. In short, it leads us to be more efficient and effective in the ways we do business.

As we discover and innovate, we must implement these innovations in the form of improved business processes. But, it takes more than just implementing the change. We must be sure that they perform. Business performance can no longer be left to chance, we must find ways to proactively manage the business in such a way that we meet the business measurements and goals. To grow, to thrive, and to survive, we must optimize.

In this chapter, we discuss the IBM Business Innovation and Optimization (BIO) initiative. We provide a general perspective on BIO and then introduce the primary strategy and components. It is a very robust initiative, addressing all the primary business areas. At a very high level, BIO includes four key approaches for getting started. These are called BIO on-ramps. For the scope of this redbook, we focus only on the *Performance Insight* on-ramp.

Performance insight is basically the process of identifying and optimizing those elements of the business that can enable an improvement in business performance. The idea here is to continually improve the business processes and enable their proactive management to insure that we meet the business measurements and goals.

The key word here is *proactive*. That is, do not wait until problems arise and then try to minimize their impact. With appropriate business intelligence and swift appropriate action, those problems can be avoided - or at least significantly minimized. And with improved and more current business intelligence and appropriate action, we can have a much better chance at achieving our business measurements.

One of the goals of any business should be to implement the capabilities for enabling the achievement of business measurement targets. And, for this, the requirement is business innovation and optimization.

There are many ways to get started on this journey. We have picked one to describe and demonstrate, and that is performance insight. We not only discuss and describe it, but we provide a case study that revolves around it. It is a fictitious case study, but it provides one simple example to illustrate how you can achieve performance insight.

Innovating and optimizing your business is a journey, and there are many paths on which you can begin this journey. Which path should you choose? Business intelligence can help in that decision. But, the important thing is to pick a path and begin.

2.1 Optimizing performance to achieve business goals

Information technology (IT) deployed in support of business, as well as government and organizations, has arrived at a natural step in the evolution toward increased automation. First steps were improved manufacturing processes and low level services support, then clearly defined functional processes within the organization (such as payroll and accounting) and discrete personal productivity applications (such as word processing, spreadsheets, and mail applications), and finally, transactions (online buying and selling, procurement, and e-Government services). Now automation is ready to move on to a higher level of functionality, but with an important caveat. That is, it is no longer sufficient to automate single processes in isolation.

For companies, this means they must integrate in better and intelligence-powered ways, that is, to continuously optimize cross-functional applications and data to have a much more variable cost structure to enable an increase in revenue and profit. As competitive pressures increase, organizations find that they need to reduce fixed costs and capital requirements and start down performance optimization paths. This is not only at the business unit level but it must be addressed as an enterprise-wide objective.

To reap the benefits of this move, in fact, to make this approach work at all, we must achieve a new level of integration among technologies and business processes. And we must take a step further. No longer can automated functions simply replace human actions, especially in the realm of decision making and judgement, but integration must include processes, technologies, and the human beings managing and acting upon them.

Today, for companies to be able to respond quickly and wisely (or at least guarantee the same level of service at a reduced budget), they must begin to change and transform their businesses in several ways. First, they must get an accurate and timely view of business conditions. Second, they need to align their strategic business goals with their underlying business processes. To support this, they must give workers access to information that allows them to act quickly and make informed decisions. Third, they must take a proactive, rather than a reactive, approach. And most important, each organization must establish a business environment that fosters and supports continual innovation and improvement.

Then in order to have better performance, it is necessary to acquire new tools to support business, as well as IT, to manage, monitor, analyze, and predict company performance. It is necessary, of course, to integrate these to optimize business performance. But, at the same time, to effectively succeed it is necessary to apply (but also expand) these tools in new, integrated, dynamic, and innovative ways.

In fact, performance optimization is a complex and articulated task that, in some cases, should be evaluated carefully and can sometimes bring you to the conclusion not to overly optimize specific aspects or processes. Modern business organizations are very dynamic and it is important to maintain a good balance between flexibility (exploration) and optimization (exploitation). This has been observed in relevant studies of the area. To see one such study, about learning systems and organizations, by Stanford professor James March, visit the Web site:

http://www.stanford.edu/group/WTO/people/board/march.shtml

A good balance is determined, in part, by how fast the environment is changing. The faster it changes, the more dangerous over-optimization (pure exploitation) becomes. Think, for example, about the traditional Telco market following the introduction of the mobile phone. A few years ago there were public phones all around the cities. Have you looked lately? That network of public phones is quickly disappearing. So, the optimization initiatives that targeted those specific business components have lost their original value. An example of such an optimization initiative could have been one to optimize the distribution of public phone appliances in a territory to match the specific phone usage and traffic in that territory. If the Telco company invested in the implementation of this sort of optimization without setting up dynamic and reusable optimization components and business processes, those investments would be lost.

Industry and domain knowledge, best practices, and services are central to the design of optimization of business performance. And, in a form more intertwined with IT than the now established emergence of a *services economy* might imply. In fact, the development of new business models, processes, strategies, and workforce management methods can themselves be viewed as comprising a series of services.

Figure 2-1 on page 13 shows a high level view of what business performance optimization (BPO) tries to leverage within a company, that is, transactional and analytical capabilities to put strategy into action. Optimizing business performance includes *closing the loop* between the operational layer of day-to-day activities, management plans, and strategic decisions. Data is collected from a number of heterogeneous sources, both structured and unstructured. And that data can be managed by applications within each line of business (LOB) and flows in a process management infrastructure.

The information management infrastructure can also help to select, transform, organize, and consolidate data on specific targets generating additional information. Information is concerned with business process data (a performance warehouse) and functional data (an enterprise warehouse). At this point, the insight process starts with the aim to elaborate information and transform it into useful knowledge for decision makers. From the strategic layer, decisions flow to management and to operations to close the loop and influence the day to day operations execution.

Optimizing business performance implies closing this loop as soon as possible by bringing to the top relevant and appropriate knowledge and, on the other side, reflecting decisions down the pyramid by influencing, as soon as possible, the ongoing execution of the business processes.

Clearly this is simply a demonstration of the flow. We understand that closing the loop involves much more, such as using measures and actions to optimize and innovate the business, for example, determining how statistics can be extracted from WebSphere Business Monitor to improve simulations, how KPIs monitored with WebSphere Business Monitor might reveal that we need to alter a business rule threshold, and so forth.

Also, this does not imply that decisions are hierarchical. They can in fact be made at any level. This simply implies a flow of data through a process to action.

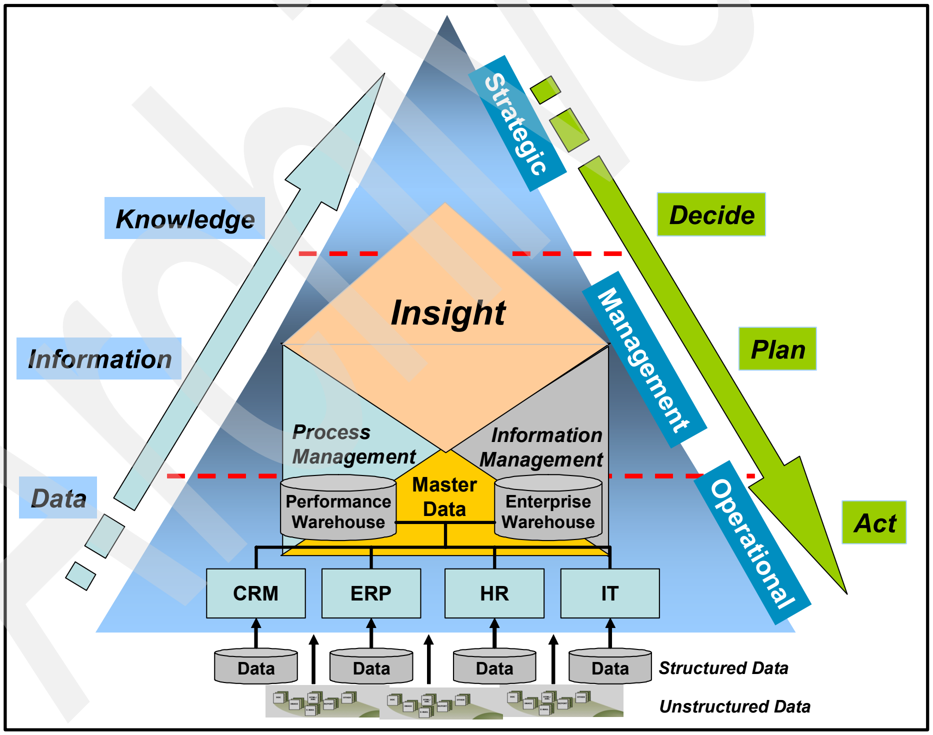


Figure 2-1 Provide Insight to decision makers and close the loop

To enable a company to set up the needed infrastructure, from the technological point of view, there are important concepts, such as virtualization and service oriented architecture (SOA), to build a flexible and reusable IT infrastructure. In fact, building an optimization infrastructure based both on virtualization and SOA concepts is an important step to preserve the investments of a company and preserve the needed flexibility to cope with change.

Other performance optimization enablers are business process management, business intelligence, and data mining.

Business process management is needed to model and run business processes in a structured and controlled way, especially those processes that cross the various lines of business and involve multiple resources and tasks (such as human and programmatic). In those processes, there can be opportunity for optimization. We also need business process management to develop a real-time performance warehouse.

Business process management does not replace existing technology investments, rather it aims to provide an environment to orchestrate and run end-to-end processes. If there is no integrated environment or business process management system, then the process activities must be managed manually. With that come inefficiency and the inability to monitor processes from an end-to-end perspective, to cope with compliance regulation problems, to generate duplications of activities, and so forth.

BI and data mining capabilities are also important enablers. They enable the search of appropriate knowledge to generate insights. These, in turn, enable you to understand the ongoing context and to match, compare, and correlate this context with the historical context, or with the market itself and to plan and take real-time optimization actions. The quantity of data and information that crosses organizations today is so huge that it cannot be managed and analyzed with traditional approaches. You need automated support for making decisions throughout the organization and to improve complementary business processes. You can optimize costs but can also increase revenue, improve product development, improve customer satisfaction, increase customer retention, attract and acquire new customers, take a continuous look at new markets to innovate, and so forth. All data for these types of mining capabilities is intrinsically cross-functional, and decision making is throughout the organization.

2.2 Layers of performance optimization

Helping companies and organizations innovate, optimize their business processes for improved performance, and increase their business value is a high priority for IBM. This can be a complex job that requires several levels of expertise and mature industrial capabilities. This is because you must deal with the complexity of the optimization problem, as well as the complexity of the company with which you are dealing.

Figure 2-2 shows the relationship between the various levels of complexity addressed by the end-to-end IBM offerings and the relative business value. This certainly does not mean the business value of Infrastructure Optimization and Virtualization is low. It simply means, from a relative perspective, that you gain even higher levels of business value as you implement the different capabilities. IBM offers services and consulting capabilities to address performance optimization across this spectrum of capabilities.

The various offering layers are incremental and start from a sound infrastructure and move on to innovative offerings that include specialized middleware, tools, and software solutions specifically for business performance optimization. They use consulting, industry specific knowledge, and best practice solution offerings and attack optimization problems from an end-to-end perspective. This involves combining IBM consulting and the power of IBM Research capabilities to provide innovation for complex business models and to solve very complex optimization problems.

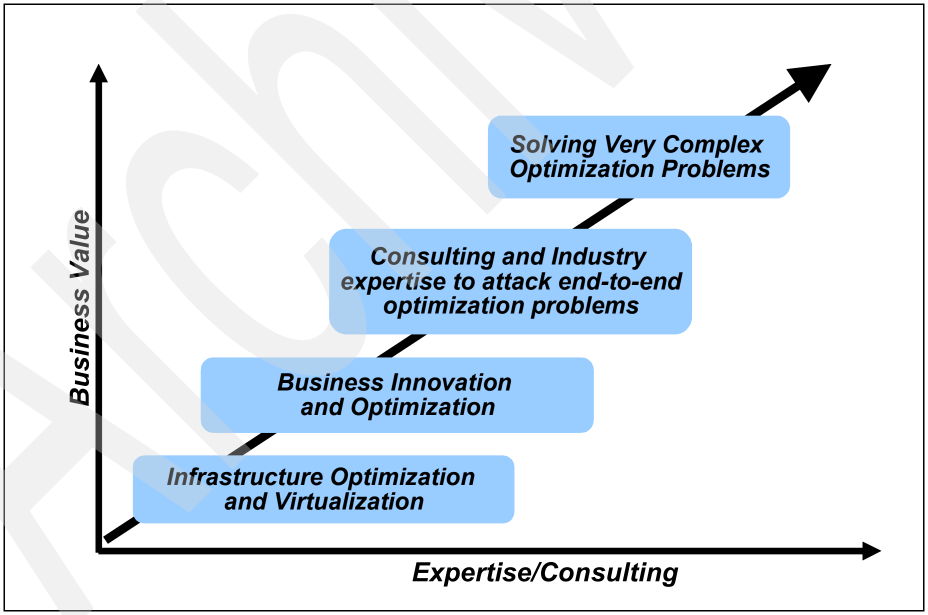


Figure 2-2 Business Performance Optimization capabilities layers

The following sections describe the layers of IBM optimization offerings and capabilities.

2.2.1 Infrastructure optimization and virtualization

Successful companies everywhere recognize that a flexible and responsive IT infrastructure plays a key role in enabling the achievement of business goals. A first step to take to optimize business performance is to optimize the IT systems' throughput and utilization. This can also play an important role in keeping the IT infrastructure simple to keep costs under control.

Owning complex infrastructures, characterized by racks of under-utilized servers, silo applications and data, disparate management systems, and manual provisioning makes IT management difficult. And this, of course, is costly and time-consuming to operate, while often limiting innovation and market-responsiveness. The need to respond quickly and effectively to the business opportunities presented by converging technologies and industries is strictly connected with the ability to manage rapidly proliferating systems, in the face of extreme budget pressures, without sacrificing performance and functionality.

IBM provides an extensive portfolio consisting of software and services as a foundation to help clients get the most out of IT. The following are three primary optimization steps:

1. Improving IT asset utilization (physical infrastructure optimization)
2. Enabling rapid response to changing business requirements (logical infrastructure optimization)
3. Employing virtualization and systems management technologies to increase flexibility and responsiveness

Improving IT asset utilization is concerned with the physical infrastructure optimization. Selecting the best architecture (for example, Blade, Cluster, and Mainframe), technologies (Industry-standard, cross-platform technologies, and the Linux® open source), and even printing solutions can lead to more cost-effective operations and infrastructure management.

Enabling the IT infrastructure to be rapid in response to changing business requirements is the next step of the optimization process. This step is concerned with a logical consolidation to make it fast and easy to allocate or add resources to meet dynamic business requirements. All IBM systems and IBM System Storage products, for example, offer the ability to either permanently or temporarily increase processor capacity, memory, or storage device utilization. IBM clients can also benefit by flexibly increasing capability models that allow, for example, the ability to scale out in a *pay as you grow* manner. So, for instance, it is possible in some hardware configurations to expand from four to thirty-two processors through affordable modular building blocks.

Other flexibility available at the logical layer is located in middleware. For example, by using IBM Grid computing, it is possible to set up a specialized form of virtualization in which server processing power, storage capacity, and network bandwidth can create a single system image that grants users and applications access to vast IT capabilities. This performance optimization gives you the power and flexibility to address new requirements, such as easily scaling out by allocating more nodes of the grid for a high-demand workload.

The last optimization step, employing virtualization and systems management technologies to increase flexibility and responsiveness, aims to transform rigid, inaccessible resources into a flexible, dynamic infrastructure that you can easily control. The aim, in this case, is to activate a virtualized environment where resources such as servers, software, and storage are pooled and shared so they can be leveraged and respond to changing requirements.

From the technical perspective once you need to deploy a new application in a virtualized environment, you look across your pooled environment and allocate available server and storage resources on existing systems. This technique is not new and has its roots in the mainframe world, where it was introduced and has been in use for over 35 years by IBM.

> **Note:** To learn more about IT Optimization, a good starting point is the following Web site:
>
> `http://www-03.ibm.com/systems/infrastructure/itoptimization.html`

2.2.2 Business Innovation and Optimization

BIO complements IT Infrastructure optimization by providing a reference architecture. Products in that reference architecture can reveal the current state of the business, enabling more informed decision making to continuously improve operations. Then leveraging technologies from business intelligence, process management, business service management, business activity monitoring, and corporate performance management can lead to continuous business performance improvement.

Innovating and optimizing gives a business the opportunity to combine market-leading software, industry expertise, and best practices for activity monitoring, process management, service management, and business intelligence with the aim of delivering an incremental approach to business improvement. In particular, you can leverage business modeling and visualize performance results in real-time dashboards.

The set of components in the reference architecture are based on a service oriented architecture (SOA) and provide the software to address the entire business performance optimization life cycle. And, it is this flexible component architecture that enables companies to preserve their corporate IT investments.

For more information about IBM BIO, see 2.3, "Business Innovation and Optimization" on page 23.

2.2.3 Consulting and integration services for optimization

Depending on which type of complexity characterizes your optimization problem, you can benefit from the help of IBM consulting and integration services. They have the consulting and the technical know-how to provide a full range of services to implement an end-to-end business performance optimization solution that addresses goals ranging from strategic to operational. For example, you need infrastructure and application expertise, but you also need the skills and abilities to work at the strategic and at the organization levels.

An optimization solution might have wide reaching, essential, impacts across the company or organization. For example, to have abilities and tools to support a company to analyze change management is a critical point for the success of the optimization work. The ability to estimate the total cost of ownership and the return on investment (ROI) of the initiative itself is important to build and update the business case concerning the optimization effects. To have the ability to decompose the project in a granular way in order to be successfully incrementally implemented and applied in practice is another example of the need to have an optimization task successfully finalized. Additionally, the specific industry optimization knowledge and best practice are important factors to consider during the optimization project realization.

IBM can mobilize the appropriately skilled resources to help improve business performance across organizational silos, across business processes, and across technology platforms. IBM Consulting, Delivery, and Integration services has packaged a number of services solutions and offerings that start from strategic consulting for optimization, and end with a full range of implementation services targeting the performance optimization task.

IBM Component Business Modeling Services, a powerful approach that allows consultants to see a business from a number of different perspectives*, can* provide a new view of the organization to create an analysis of business components that cross independent business units.

Business components represent all of the unique activities that an organization pursues to create value. To create each business component, all people, processes, and technologies that enable these components are considered, acting as stand-alone building blocks that deliver value to the organization. A business offers goods or services, and so must a business component if it is to operate independently. In turn, a business component uses goods and services provided by other components and external suppliers. Business components offer business services to other business components and external parties. In summary, view an enterprise as a network of semi-independent business components, each of which uses business services which the others provide. Value to an external customer is provided by networks of cooperating business components.

One of the primary activities of the *IBM Component Business Modeling Services* is to draw a component map, which is essentially a view of the business that allows you to understand aspects of your company such as which parts of a business are differentiating, how resources are being consumed, and how effectively the company business and IT strategies are aligned. The IBM approach helps clients develop a greater understanding of their enterprise business models and an improved ability to prioritize what needs to be done to become optimized.

> **Note:** For more information about IBM Services offerings for performance optimization, see the following Web site:
>
> `http://www-1.ibm.com/services/us/bcs/html/bcs_makesusdifferent.html`

2.2.4 Solving very complex optimization problems

Solving complex business problems often amounts to having just the right mix of business and technical knowledge and expertise. For this reason, IBM recently started delivering new services capabilities centered around its world-renowned research and consulting strength, the On Demand Innovation Services (ODIS), and IBM also created innovative specialized service centers to deliver complex end-to-end business optimization solutions.

Within the ODIS, IBM commits to working directly with clients to develop innovative solutions that yield solid business results putting together IBM Research and IBM Business Consulting Services. ODIS assembles teams of experts comprised of business consultants, mathematicians, and scientists, to offer clients breakthrough approaches to bringing their ideas to life, finding new opportunities in existing marketplaces, and developing game-changing projects to move ahead of their competition.

To better match those capabilities with clients' needs, ODIS solutions have been organized into cross-industry business areas and specialized initiatives. One such initiative is specifically concerned with *Business Optimization and Analytics* that aims to optimize, plan, model, analyze, and transform businesses to be an

On Demand Business. Another is the *Center for Business Optimization* offering initiative, in which Business Optimization and Analytics expertise constitutes the basis for the offering. The Center for Business Optimization leverages IBM's global resources and capabilities, which include:

- Business consultants with the experience and expertise to improve business performance. They service key industries and businesses in virtually every country.
- Mathematicians and operations research specialists from IBM Research with expertise in data mining, optimization, simulation, and statistical modeling.
- Advanced technology and services — from deep computing power to hosted offerings in which IBM performs the analytics for you.
- Business Partner relationships with major industry application software vendors.
- Collaborative arrangements with leading universities and operations research organizations.

Many of the Center solutions also leverage the IBM open, scalable, security-rich platform for On Demand Business and tap into the computing power of the IBM global network of supercomputing centers.

The Business Optimization and Analytics initiative applies cutting-edge models, algorithms, software assets, and expertise to help clients quickly and accurately solve complex optimization problems. Business Optimization and Analytics provides solutions that help companies improve speed, accuracy, and quality in resource planning and scheduling problems whether they arise in long-range strategic plans, day-to-day tactical operations, or anywhere in between.

The area includes several optimization specialties, such as large-scale optimization, data mining for business intelligence, resource allocation in uncertain environments, risk management optimization, complex supply chain optimization, marketing investment optimization, and dynamic pricing optimization.

Those specialties focus on helping business and government organizations to reduce cost, mitigate risk, and maximize return on investment. For example, for an electric utility this might mean accurately forecasting demand to efficiently scheduling power generators so that requirements are met at a minimal cost. For an investment group, this might mean identifying a range of optimal portfolio allocation strategies for improved asset liability management for a manufacturing company, which can mean better gauging tool capacity to meet uncertain future demands. And for a transportation company, this means improving fleet utilization while maintaining, for example, its 99% on time pickup record by using an optimization system capable of optimizing driver schedules and responsiveness to changing conditions.

The following is a sample list of optimization specialities:

- Large-scale optimization consists of a growing collection of innovative software and leading-edge expertise that helps solve large and complex optimization problems in areas such as planning, scheduling, and pricing.
- Resource allocation in uncertain environments is concerned with the development of tools that allocate resources to minimize cost or maximize return in uncertain and dynamic environments. Unlike traditional resource allocation tools, which develop projections based on historical relationships, these tools also take into account the random events and changes that can occur.
- Risk management optimization provides tools to identifying potential fraud, evasion, and abuse in health care, tax reporting, and customs. Risk management optimization can provide significant improvements in detecting and investigating fraudulent behavior through advanced data mining for business intelligence that leverages the IBM Research project, Data Analytics Research (DAR).
- Complex supply chain optimization provides systems able to plan optimal inventory levels based upon manufacturing and warehousing requirements. In particular, several optimization abilities are available, such as reducing inventory while maintaining, or even enhancing, service levels — often in the face of sporadic or extreme demand patterns; determining optimal buffer stock levels, reorder points, and production batch sizes for every item in a warehouse; developing optimal production design and operations scheduling of materials by integrating customer orders with manufacturing production capacity and capabilities.
- Marketing investment optimization is dedicated to uncovering opportunities to maximize the impact of marketing spend. The specific optimization focus in this speciality is to provide tools to the marketing departments which are being asked to show an ROI for the money they spend. In particular, marketers need to understand the impact that marketing activities can have on sales, the propensity to buy at the time an offer is made, and the overall customer relationship. Business optimization algorithms and predictive modeling techniques can help determine an optimal interaction strategy for each of your clients over the client life cycle. These solutions, which can be integrated into existing customer relationship management (CRM) systems, provide marketers with a framework for planning and budgeting targeted marketing actions that can help maximize return on investment and customer loyalty.

- Dynamic pricing optimization addresses complexities of establishing pricing strategies. In particular, using strategic pricing solutions, IBM consultants can help improve your pricing and contracting processes and mathematically model your company environment to drive unique insights into price elasticity, product affinities, promotional drivers, and other variables that impact demand.

Solutions are built on common platforms that are independent of applications, yet leverage existing infrastructures.

> **Note:** To learn more about ODIS capabilities and the Center for Business Optimization initiative, see respectively these Web sites:
>
> `http://domino.research.ibm.com/odis/odis.nsf/pages/index.html`
>
> `http://www-1.ibm.com/services/us/index.wss/bus_serv/bcs/a1008891`

2.2.5 The Services Sciences, Management and Engineering

One of the IBM medium and long term strategies and research initiatives able to influence Business Performance Optimization is the *Services Sciences, Management and Engineering* initiative (SSME). This is a strategic initiative to bring together ongoing work in computer science, operations research, industrial engineering, business strategy, management sciences, social and cognitive sciences, and legal sciences to develop the skills required in a services-led economy.

The world economy is experiencing the largest labor force migration in history, driven by an environment that includes global communications, business growth, and technology innovation. For instance, investigations by analysts have estimated that services now accounts for more than 50% of the labor force in Brazil, Russia, Japan, and Germany, as well as 75% of the labor force in the United States and the United Kingdom.

This unparalleled segment growth is changing the way companies organize themselves and optimize their performance, creating a ripple effect in industries and universities that are closely tied to these organizations. For instance, historically, most scientific research has been geared to supporting and assisting manufacturing, which was once a dominant force in the world economy. Now that economies are shifting, industrial and academic research facilities need to apply more scientific rigor to the practices of services, such as finding better ways to use mathematical optimization to increase productivity and efficiency on demand.

Unfortunately, this shift to focusing on services has created a skills gap, especially in the area of high value services, which requires people who are knowledgeable about business and information technology, as well as the human factors that go into a successful services operation. For example, this skills gap, from a business intelligence perspective, is one of the inhibiting factors that, during the last few years, delayed the effective usage of solutions within the company context.

There are several other connections between BPO and SSME. In fact, BPO is a reasonably complex task that, in most cases, should be evaluated carefully and bring you to the conclusion to not overly optimize specific aspects or processes. Modern business environments are dynamic and it is important to maintain a balance between flexibility (exploration) and optimization (exploitation). The right balance is determined in part by how fast the environment is changing. The faster the environment changes, the more dangerous over-optimization (pure exploitation) becomes.

Service systems occur at many levels, such as global, national, industry, firm or enterprise, work system, and knowledge-worker level (or professional level). Optimization at one level can cause challenges for the level above or below. Over-optimization can create challenges in a dynamic environment.

One of the grand challenges of SSME is how to invest in a service system for optimal overall performance in service productivity and service quality across time.

Many leading universities have begun investing in this area, working in tandem with thought leaders in the business world, aimed at preparing students for real-life services practice. BPO has become a primary focus.

> **Note:** To learn more about the SSME initiative, see the Web site:
>
> `http://www.research.ibm.com/ssme`

2.3 Business Innovation and Optimization

BIO enables you to see in real time the current state of company business, enabling you to make more informed decisions, be more predictive, and continuously improve operations. It leverages technologies from business intelligence, process management, business service management, business activity monitoring, and corporate performance management to enable continuous business performance improvement.

The focus on continuous business performance improvement plays a central role in the IBM approach and, at the same time, differentiates it as the state-of-the-art in this field. In fact, the ability to stay in touch with market and customer demands becomes increasingly challenging. Globalization forces companies to quickly adapt to new markets, evolving employee skills, and new competitors. Significant changes to business processes that were once made annually by organizations are now being made monthly or even weekly. In addition, a renewed emphasis on growth forces companies to find new ways to outwit and outmaneuver their competition. Yet at the same time, their continuing focus on cost containment requires them to invest more prudently than in the past.

Clearly, market conditions are constantly changing; to survive, a company needs to become more responsive to the fluid conditions that create performance challenges. It is necessary to develop the flexibility to meet shifting customer demands, to respond to new market opportunities, or to be prepared for any new external threat. In short, it is necessary to *continually innovate,* optimize, and change as rapidly as the business environment demands.

Continuous innovation aims to develop appropriate sensors and tools, from the IT perspective, that are able to capture data as soon as possible and supply the ability to adapt the company or the organization to changing market dynamics and everyday operational disruptions in a way that reduces costs and generates competitive advantage. To achieve this state, it is important to align strategic and operational objectives with business activities to fully manage performance. The additional insight into the performance of company processes allows you to see, for example, those processes that have a high productivity and those that are experiencing problems, enabling management to continuously refine and innovate.

In pragmatic terms, BIO can help you by:

- Supporting continuous innovation and improvement. You can establish a flexible, readily adaptable business environment that provides ongoing performance enhancements and optimization.
- Enabling more informed and effective decisions. You can optimize decision making with real-time contextual insight, resulting in faster, more appropriate actions.
- Facilitating the alignment of business objectives. You can determine and understand goals from strategy to operations, align measures and orchestrate actions across business and IT.
- Helping manage operational disruptions effectively. You can better anticipate disruptions in day-to-day business operations and quickly take direct, proactive actions to improve results.

2.3.1 An approach to BIO

IBM BIO helps you to see deep into your business, giving you insight into vital details about your operations and processes to enable continuous business performance improvement. With this approach, you are able to follow a straightforward path to improve responsiveness, lessen the impact of potential problems, quickly capitalize on strengths, and refine processes as needed to meet changing requirements.

The BIO approach to business performance improvement is incremental, allowing you to implement your own approach in phases and to leverage your existing investments. You can address performance needs at the pace your business requires, while focusing on specific initiatives to realize rapid, targeted results. The most common types of initiatives that stimulate the need for business performance improvement are risk management, regulatory compliance, asset utilization, growth, customer interaction, and cost reduction.

Key to this approach are IBM and IBM Business Partner software offerings that use service oriented architectures (SOAs). A SOA segments applications into individual business functions and processes called *services*. Because they are flexible, extensible, and open standards-based, these advanced software offerings allow you to build, deploy, and integrate services to meet evolving business process needs while leveraging existing investments.

A model for continuous business performance improvement is represented by the SOA life cycle activities in Figure 2-3 on page 26. However, the starting points and sequence of the activities can vary depending on whether your needs are for process management, activity monitoring, or real-time analysis.

With the IBM BIO approach, you can gain the benefit of software, industry expertise, and best practices at each key stage of the life cycle. The life cycle reflects the SOA general foundation and goes through the four steps of the SOA, which are Model, Assemble, Deploy, and Manage. Additionally, there is an initial step which refers to the optimization problem assessment itself, which is Governance and Processes. These steps are expanded in the following sections.

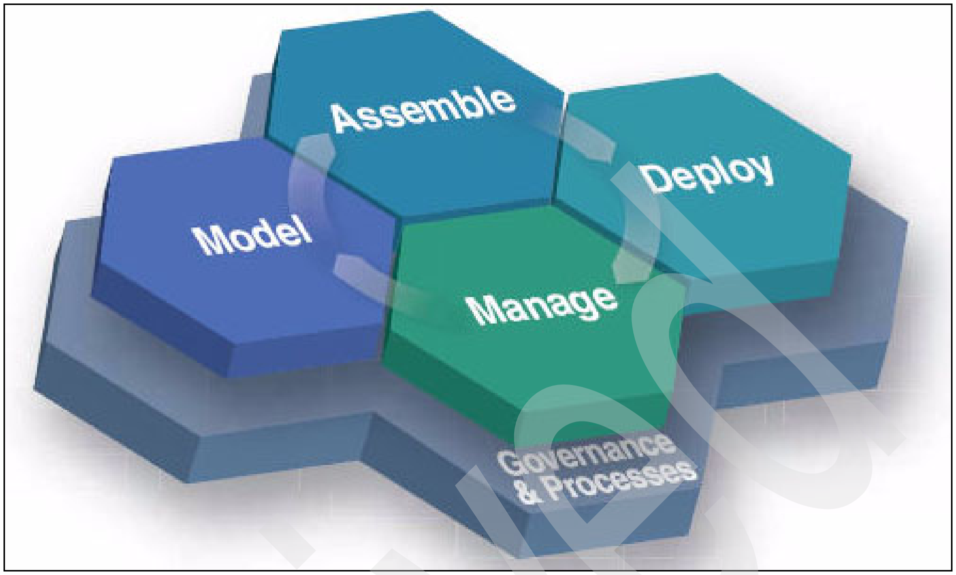


Figure 2-3 The SOA life cycle

Governance and processes

Improvement starts with the identify action stage of business goals. At this stage, a key piece of IBM technology comes into play. It is the IBM Component Business Modeling Tool that, with help from IBM consultants, provides a means to identify and understand your key business goals and associated processes. By using this tool, consultants help you gain significant new insights into the strategy, technology, operations, and investment alignment of your organization.

The analysis starts with a complete model of the essential business processes in your industry, using it to identify differentiating and non-differentiating components and to isolate those components presenting immediate opportunities for growth, innovation, or improvement. This remarkably efficient discovery process helps the company prioritize initiatives and ensure that operational and capital expenses are aligned with the company's overall business strategy.

Model

Business modeling is a key capability that helps capture critical business processes, such as business policies, key performance indicators (KPIs), business events, and situations, and the actions necessary to respond to events and optimize performance. This phase is realized by a business analyst jointly with line-of-business managers and process developers.

The business models serve as the basis for the instrumentation of business processes for monitoring and optimizing business operations. Monitoring enables visibility into business performance problems on a real-time basis and provides the ability to take timely actions to optimize business operations. In other terms, by using modeling, you can simulate a process to understand how it will perform before implementation and define key performance indicators to monitor the success and efficiency of the process. Modeling allows you to align objectives and priorities between IT and business.

This BIO focus area ties into the current industry direction of Model Driven Development (MDD) well. For example, the business process is modeled by a business analyst and then utilized by an IT department to modify the same model to deploy into an IT environment.

IT models help capture the managed IT resources and services within an organization. The topology of resources, such as hardware systems, software infrastructure, and the properties and relationships across these resources, can be reflected in models.

Capturing models of IT resources can offer significant benefits to IT management activities, from service level agreement (SLA) enforcement to transaction performance monitoring and problem determination. IT models can help with causal analysis by correlating events back to the resources that directly or indirectly caused them to occur. Also, IT models can help in change management, which is a major issue for many companies.

To align IT objectives and performance with business needs, IT models should be correlated to business models. Modeled relationships between the elements in a business model (processes, organizations, and services) and the IT resources on which the business depends (systems and software) can be used to provide shared context and can help ensure the health and availability of IT infrastructures for optimal business performance.

From a technical perspective, a model is similar to a flow chart. Figure 2-4 on page 28 shows a portion of the process model built during this redbook case study. This project was named the Good Return Process and was produced using the WebSphere Business Modeler.

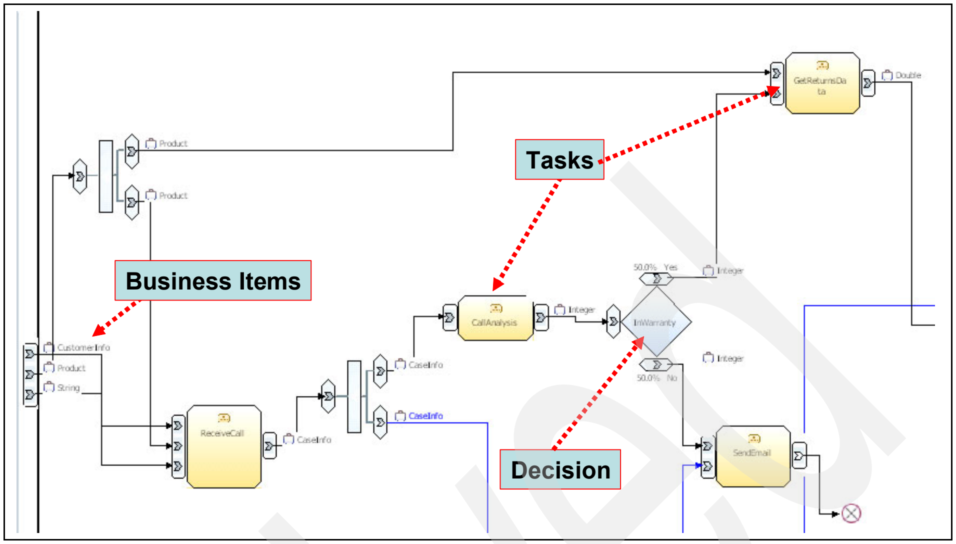


Figure 2-4 An example of a Process Model

A *process model* consists of a sequence of activities included in the modeled business process that can be performed either by human beings or automated systems. You can also arrange process activities into a hierarchy, activating a task or a sub-process in a specific activity. Each task performs some function or activity. Tasks are the basic building blocks of a process diagram. You can specify business items (data) that travel from one activity to another and are produced from one activity, stored in a local repository, and then consumed by others from that repository. You can define decision points with alternate paths to follow.

Once the process model is ready, you can simulate it by attaching the estimated duration and the cost of the resources consumed during each activity to each activity and to the output of conditional branches. When designing a model, it is best to think in business terms, leaving implementation details for the following phases.

Although this is an initial step, you can still meet optimization goals. In fact, you can model the current process (the *as-is*), model alternatives (the *to-be*), assign costs and evaluate them by using simulation, and then select the model that best fits your specific situation.

Assemble

After the process has been optimized, you can use development software to assemble assets necessary to support the process. In this step, the IBM BIO reference architecture is supported by a specific tool, which is WebSphere Integration Developer.

In particular, once the business users have documented the strengths and weaknesses of the process, the model can be passed to the IT community for a variety of implementation scenarios. The model does not change, but the data associated with the model is expanded to add technical details. The model generates a skeleton process that is useful to connect business analysis steps and their corresponding implementation.

Each activity in the process is associated with a software component to provide the implementation. The assembly process is based on a SOA. The primary objective of this approach is to no longer have IT solutions built from the ground up, with all needed application services developed with the solution. Rather, a software component can access external reusable services through integration adapters. Existing IT systems can be wrapped in a way that they can appear as services providers to integration adapters.

The result of an assemble activity is an implementation of the business process. The designer, starting from the modeled process generated from the WebSphere Business Modeler, has the responsibility to assemble intra-enterprise and inter-enterprise services into the business processes.

Figure 2-5 on page 30 shows the business process diagram for a process we implemented during the redbook case study, the Good Return Process. It was produced by WebSphere Integration Developer after importing the model designed using WebSphere Business Modeler. In particular, this figure shows one of the diagrams used by integration developers to specify the process task's logic. Programming tools have been designed so that users can easily compose integrated business solutions without programming skills. To this end, you can easily work with the business process in an intuitive graphical programming environment called the *process editor* and then deploy it to a runtime environment for execution.

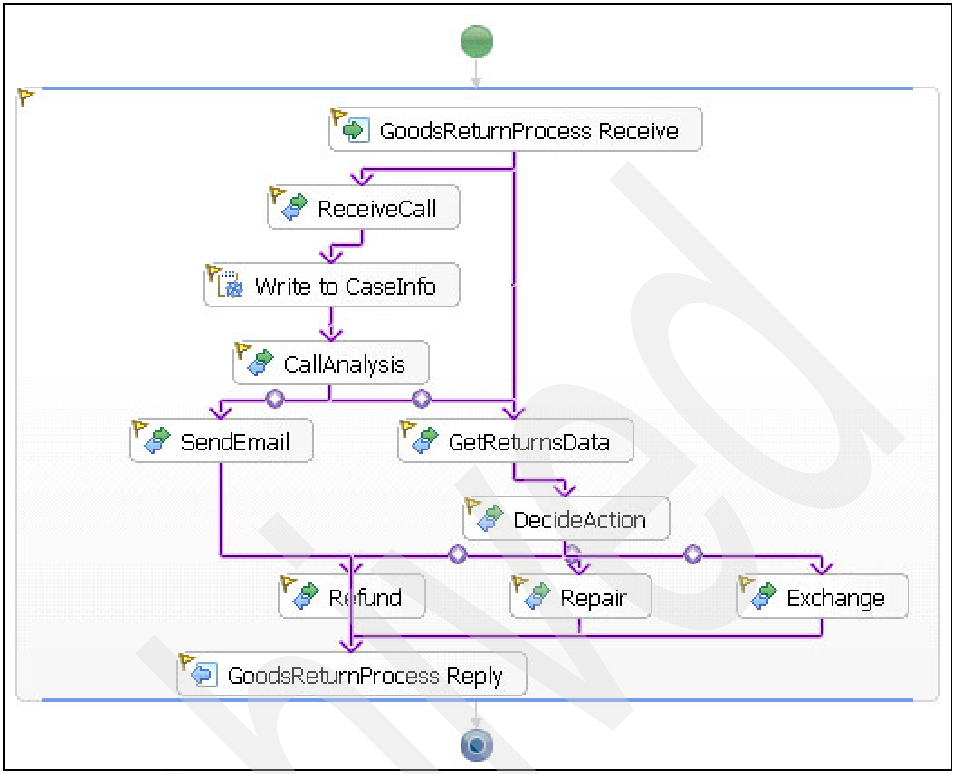


Figure 2-5 The business process diagram

Figure 2-6 on page 31 depicts another essential diagram used in this phase, the assembly diagram. In this case, two processes were built during the redbook case study, the Good Return Process and the Problem Evaluation Process. The business process model designed and finalized in the previous step is now managed by
a developer that assembles components. The various tasks included in the business model are seen by the integration developer as components that should be locally implemented (for example, as a Java module or a human task) or connected to external services.

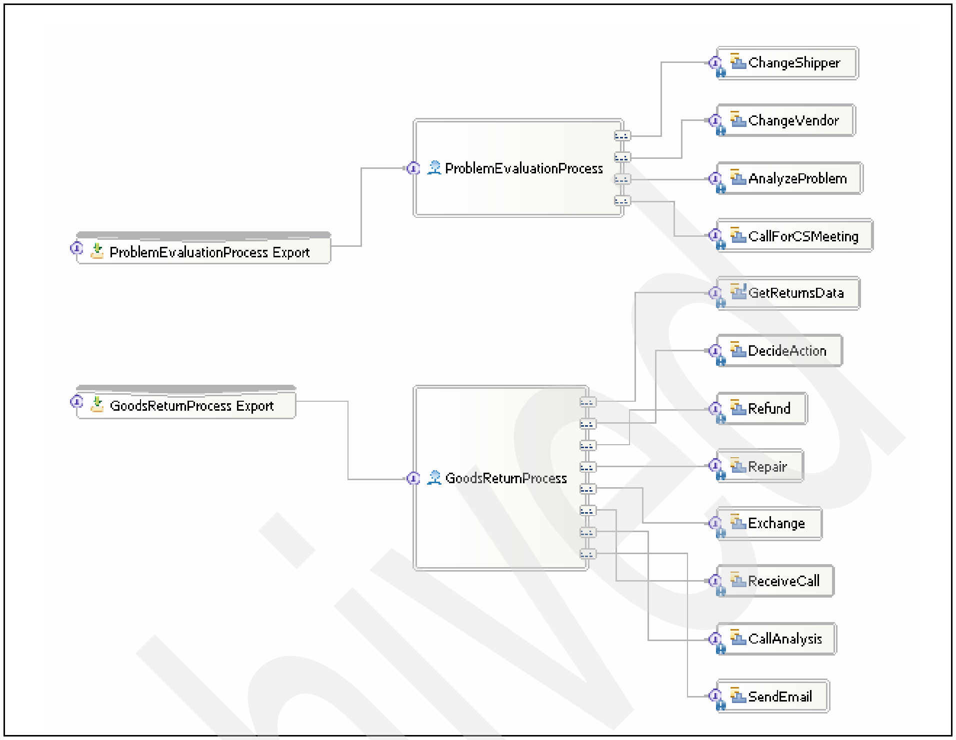


Figure 2-6 An example of the result of the Assembly phase of a business process

Deploy

Next, you can deploy the processes, policies, and resources that you have modeled and assembled. These can include data models or models for strategic or business operations.

The deploy activity transforms, packages, distributes, and installs the implementation of the models created during modeling. It converts platform-independent models to technology-specific implementations. The deploy activity must be capable of handling virtualized resources, such as virtual teams for human resources and workload management for infrastructure resources. It should also support the deployment of new components into an existing deployed system (*dynamic redeploy*) and into a currently executing solution (*hot deploy*).

Deploying the process delivers tools to integrate and manage the business processes. The use of technologies, such as information integration and business intelligence, specifies the interfaces used to access and analyze the information that flows from all the other information sources in an enterprise for dynamically controlling business processes and managing business and IT performance. The common event infrastructure provides a common architecture for creating, transmitting, and distributing business process events and IT events.

At this stage, processes are deployed into a *process engine* to be executed. The process engine has the responsibility to execute each instance of the process and appropriately route every activity included in the process.

Human activities are managed by using a workflow approach. In this case, notifications about work items are generated and delivered to the specific receiver for appropriate action. The flow is constantly monitored and, in each instant, the process server knows the status of each work item and the queues associated with each activity and takes escalation actions. Automated activities can involve the integration with external systems for both short and long running transactions. One of the main roles played in this case by the process engine is to orchestrate mapping data between the various systems involved and also to store the history of each running process.

Manage

After deploying a new process, you embark on the final, and probably most important, stage, which is management. Dashboards, KPIs, and alerts tied to real-time event-based data helps users monitor and manage process performance. You can analyze your progress with a process and use this information for continuous improvement to your business. The analyze activity is used by the monitor activity to calculate predefined KPIs and perform *ad hoc* or impromptu analyses. These KPIs and analyses can be used with associated historical data to evaluate the performance of the organization. Analysis of the information is provided with context to the users who make the decisions on process metrics to detect anomalous situations, understand cause and effect, and take action to maintain alignment of processes with business goals.

In Figure 2-7 on page 33, we show an example of a dashboard taken from a demonstration of an insurance application. The dashboard gives management a number of strategic elements that require focus and monitoring. For example, it shows new business growth by category and an overview financial snapshot. Management can monitor these elements to make sure they are in line with the business goals and strategy. If not, action can immediately be taken. The dashboard can also be used for management by exception, such as identifying deviations from the norm.

The dashboard also gives current status on a number of projects, with appropriate alerts. Again, management can focus their attention because they now know where it is required.

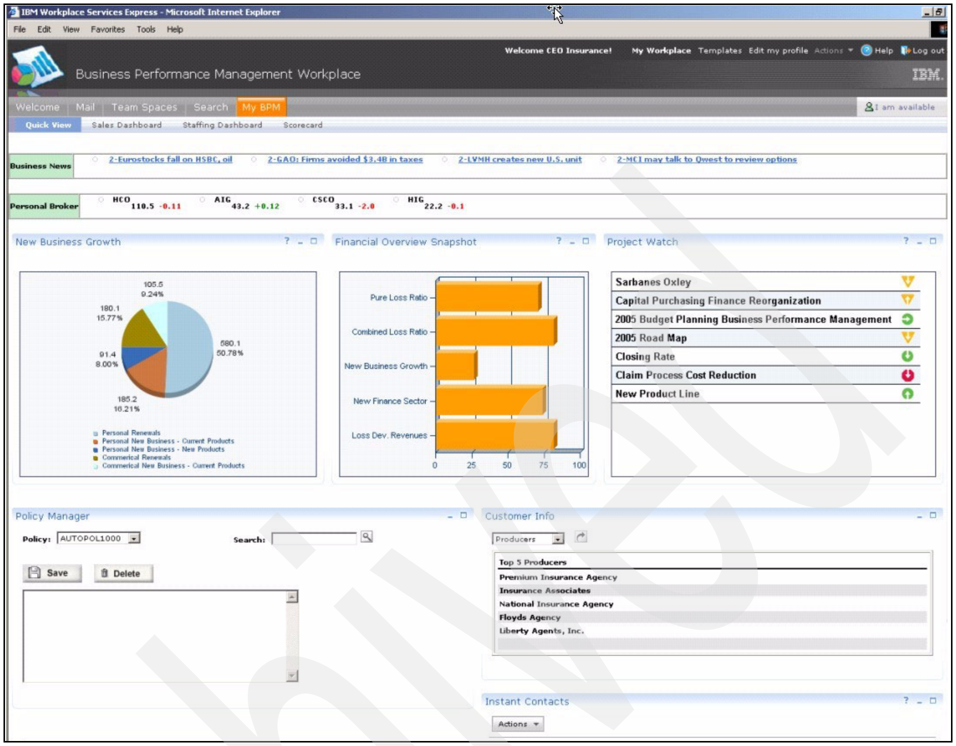


Figure 2-7 An example of a dashboard for an insurance company

In Figure 2-8 on page 34, we show an example retail dashboard from another demonstration. This dashboard shows a list of the key business areas and the appropriate alert status. There is also a summary of the current business financial status. With this type of information, management could have the capability to both monitor and impact the measurements.

It is also important to understand that statistics from WebSphere Business Monitor can be imported back into WebSphere Business Modeler to improve the simulations and analysis with real data.

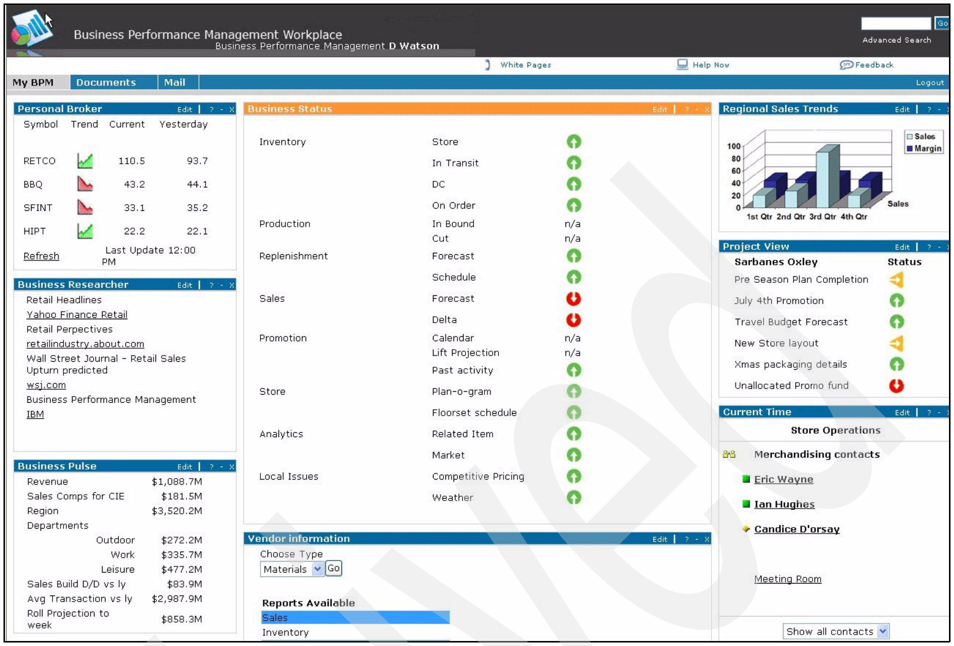


Figure 2-8 An example of a dashboard for a retail company

In general, a key technology for monitoring and analyzing business performance is data warehousing. A *data warehouse* brings together data from multiple systems, both inside and outside of the organization. A data warehouse also provides access to both summary data and detailed business transaction data. This data can be historical in nature or can reflect a “close to real-time” status of the business operations. The availability of detailed data enables users to drill down and perform in-depth analyses of business performance.

Of course, business intelligence applications and tools play an important role for analyzing performance data and historical data in a data warehouse. Analysis of business event data and other historical business data is necessary to diagnose business performance problems. It is also crucial for evaluating decision alternatives and planning appropriate corrective actions when business performance management issues are detected. The analyze activity supports business services management by predicting potential IT infrastructure problems before they occur. Analyzing historical information about the health and performance of the infrastructure can help predict potential violations of service level agreements or internal resource performance thresholds before they actually materialize. Analysis using ad hoc methods means that the monitoring capability of business process management must be designed in such a way that it remains dynamic and easily changed.

Businesses constantly want to view information about performance in new ways that are more informative and more easily understood. And IT must be positioned to provide that level of support.

2.3.2 A software component platform for BIO

From a technology perspective, a BIO platform can offer a tightly integrated, but loosely coupled, component architecture based on a service oriented architecture. Figure 2-9 shows a reference architecture in which all types of application services support a SOA-based business context.

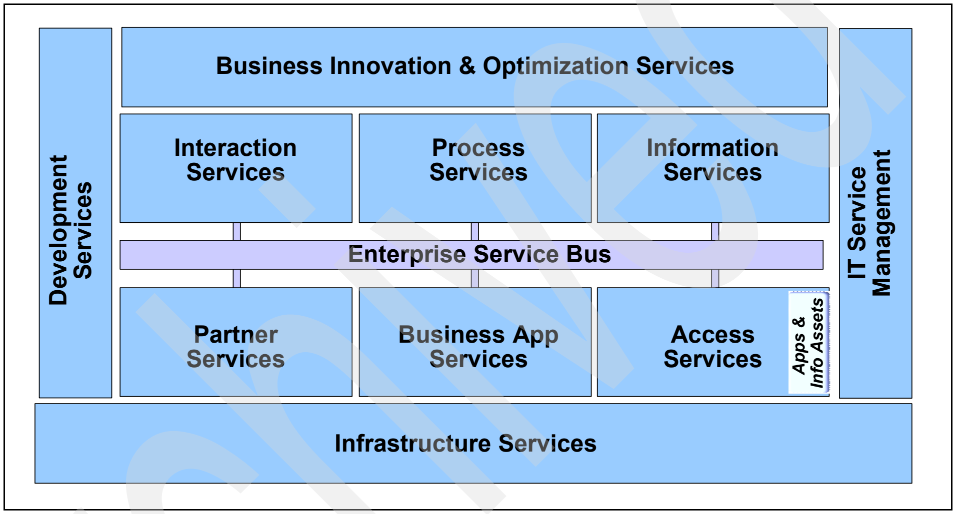


Figure 2-9 SOA reference architecture

The SOA reference architecture is a complete and comprehensive architecture that covers all the integration needs of an enterprise. Its services are well integrated and are delivered in a modular fashion, allowing SOA implementations to start at a small project level. As each additional project is addressed, new functions can be easily added, incrementally enhancing the scope of integration across the enterprise. In addition to supporting SOA strategies and solutions, the architecture itself is designed using principles of service orientation and function isolation.

SOA is an integration architecture approach based on the concept of a service. The business and infrastructure functions that are required to build distributed systems are provided as services that, collectively or individually, deliver application functionality to either user applications or other services. It specifies that, within any given architecture, there should be a consistent mechanism for services to communicate. That mechanism should be loosely coupled and support the use of explicit interfaces.

A SOA can bring the benefits of loose coupling and encapsulation to integration at the enterprise level. It applies successful concepts proven by object-oriented development, such as component-based design and enterprise application integration technology, to an architectural approach for IT system integration.

Services are the building blocks of SOA, providing function from which distributed systems can be built. Services can be invoked independently by either external or internal service consumers to process simple functions or can be chained together to form more complex functionality and, therefore, to quickly devise new functionality.

By adopting a SOA approach for BIO-related applications, companies can build flexible systems that can quickly implement changing business processes and can make extensive use of reusable components.

> **Note:** For more information about SOA, visit the following Web site:
>
> http://www-306.ibm.com/software/solutions/soa

BIO-related services could incorporate monitoring capabilities that aggregate operational and process metrics in order to efficiently manage systems and processes. Managing these systems requires a set of capabilities that span the needs of IT operations professionals and business analysts who manage the business operations of the enterprise. These capabilities are delivered through a set of comprehensive services that collects and presents both IT and process-level data, allowing, for example, business dashboards, administrative dashboards, and other IT level displays to be used to manage system resources and business processes.

Through these displays and services, it is possible for business IT personnel to collaborate to determine, for example, which business process paths might not be performing at maximum efficiency, the impact of system problems on specific processes, or the relationship of system performance to business process performance. This collaboration allows IT personnel and assets to be tied more directly to the business success of the enterprise than they have traditionally been.

From a technological point of view, the BIO reference architecture includes a set of software components that can deliver key business performance functions for modeling and implementing processes, monitoring performance, and optimizing business processes and performance. A high-level functional component model diagram to set up BIO-related application services is presented in Figure 2-10.

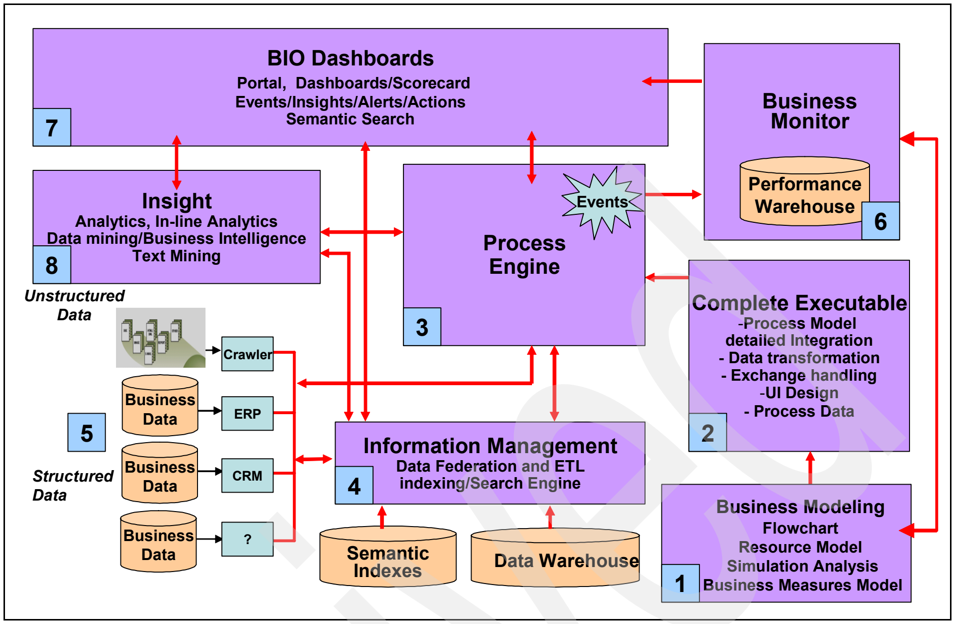


Figure 2-10 A functional component diagram of BIO services

A typical BIO-related application life cycle starts with understanding the strategy, goals, success factors, and requirements. Then, the process modeling phase can begin. In Figure 2-10, this is represented by component number 1. The modeling provides initial understanding of the enterprise processes. During this phase, essential data and artifact modeling, organization modeling, resource modeling, time line and location modeling, and business process analyses are realized. In this phase are defined KPIs, and a first degree of optimization is obtained by using simulation abilities. Process statistics can also be fed back from WebSphere Business Monitor to WebSphere Business Modeler for further simulation and analysis. Also implied here is the deployment of the business measures model from WebSphere Business Modeler to WebSphere Business Monitor.

At the next activity (component 2), an executable process is developed and fully implemented. The implemented process is then deployed into a process engine (component 3) that can run each process activity and orchestrate appropriately the execution of each activity, for both human tasks and automated functions.

The process engine is at the center of the figure and can access and integrate already available IT systems (component 5). Through an information management component (component 4), the process engine acquires information content from the various systems (component 5). Alternatively, at runtime, some process activities can supply information to other systems and contribute data warehouse data and semantic indexes for unstructured data by using analytics, other business intelligence, and data mining components (component 8).

During the process execution, events that are generated flow into the monitoring component (component 6). There is also information sent back to WebSphere Business Modeler from WebSphere Business Monitor, such as process statistics, to affect a closed loop environment. For example, process statistics can be imported into WebSphere Business Modeler to improve further simulation and analysis. In general, BIO application components (component 7) run within a Portal, and they can benefit from services supplied by the monitoring component through dashboards or interact with the process engine services. A BIO-related application can, of course, interact directly with the information management component (component 4) or require services of the inline business intelligence and data mining components (component 8).

2.3.3 Mapping BIO functionality and IBM products

IBM offers a comprehensive and advanced product suite to meet the requirements for building BIO-related application scenarios, including the component model. In particular, each BIO function is mapped with one or more IBM products, and all are fully integrated.

In Table 2-1 on page 39, for each BIO function, we list possible IBM product and middleware combinations that can be used for that component.

Note that some products listed are really comprehensive platforms that aggregate various products in a specialized bundle, such as the DB2 Data Warehouse Edition that includes, for example, DB2 Alphablox for inline analytics and WebSphere Information Integrator Standard Edition.

Table 2-1 BIO functions and IBM product mapping

Function	IBM product mapping
Business Modeling	WebSphere Business Modeler
Process Assembly	WebSphere Integration Developer (Process Assembly), IBM Rational Application Developer, IBM Rational Software Architect (component construction)
Process Engine	WebSphere Process Server
Information Management and Integration	IBM WebSphere Information Integrator (Information consolidation, search, and indexing)
Business Monitoring	WebSphere Business Monitor (Business Monitoring), IBM Tivoli Business Systems Manager, Tivoli Service Level Advisor (Business Systems Monitoring in the context of IT)
Business Intelligence	DB2 Data Warehouse Edition (Data Warehouse, data analytics, inline analytics)
Dashboards	IBM WebSphere Portal, IBM DB2 Alphablox, WebSphere Business Monitor

In the following sections, we provide brief descriptions of the products listed in Table 2-1. We include a link to the IBM Web site with additional product information for each product.

In Chapter 6, “Case study software components” on page 123, we describe in more detail all of the products used to set up the BIO application case study developed in writing this redbook.

IBM WebSphere Business Advanced Modeler

http://www-306.ibm.com/software/integration/wbimodeler/advanced/

This product bridges the gap between business and IT. It is based on Eclipse technology, providing robust functionality for process modeling, enterprise modeling, essential data and artifact modeling, organization modeling, resource modeling, time line and location modeling, simulation, and business process analysis.

IBM WebSphere Integration Developer

http://www-306.ibm.com/software/integration/wid/

This product enables integration developers to assemble complex business solutions using minimal skills. Based on Eclipse technology, WebSphere

Integration Developer is a tool that helps you rapidly assemble business solutions that describe all styles of processes with one programming model based on Business Process Execution Language (BPEL). It also enables Business-Driven Development, fully integrating with WebSphere Business Modeler to import models for rapid implementation. And it delivers constructs for dynamic processes, including business rules, business-state machines and selectors, events, and role-based tasks capabilities. It provides a single tool to describe all of your processes based on standards and deploys them on WebSphere Process Server. It also integrates testing, debugging, and deployment for solution development to WebSphere Process Server and WebSphere Enterprise Service Bus.

IBM Rational Application Developer

http://www-306.ibm.com/software/awdtools/developer/application/index.html

With this comprehensive IDE product, we can quickly design, develop, analyze, test, profile, and deploy Web services, Java™, J2EE, and portal applications. Optimized for IBM WebSphere software, and supporting multi-vendor runtime environments, IBM Rational Application Developer for WebSphere Software is powered by the Eclipse open source platform so developers can adapt and extend their development environment to match their needs and increase their productivity.

IBM Rational Software Architect

http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html

With Rational Software Architect, you can unify all aspects of software design and development and develop applications and Web services more productively than ever by exploiting the latest in modeling language technology. You can also review and control the structure of your Java and service-oriented applications, leverage an open and extensible modeling platform, simplify your design and development tool solution, and integrate with other facets of the life cycle.

IBM WebSphere Process Server

http://www-306.ibm.com/software/integration/wps/

WebSphere Process Server is a robust process automation product with advanced human workflow, business rules, application to application (A2A), and business to business (B2B) capabilities, all on a common, integrated SOA platform with native Java Message Service (JMS) support. WebSphere Process Server now is built on WebSphere Enterprise Service Bus (ESB) to provide a standards-based integration platform that facilitates connectivity between services.

Concerning business processes, this product implements a WebSphere-BPEL-compliant process engine. It represents the fourth release of a business process choreography engine on top of the highly scalable WebSphere Application Server. WebSphere-BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. Support for WebSphere-BPEL includes:

- Advanced process editing modality through the WebSphere Integration Developer component
- A visual business process debugger to step through and debug WebSphere-BPEL business processes
- Long- and short-running business processes compensation support to provide transaction *rollback*-like support for loosely coupled business processes that cannot be undone automatically by the application server. Compensation enables undoing already committed transactions in case of a failure in the business process.

IBM WebSphere Business Monitor

http://www-306.ibm.com/software/integration/wbimonitor/

This product enables you to monitor business processes in real time, providing a visual display of business process status. In particular, it is able to generate alerts and notifications to key users to facilitate continuous improvement of your business processes. It provides a customizable dashboard, implemented as WebSphere Portal pages, that is visually intuitive and features scorecards, key performance indicators, and gauges. The dashboard supports multidimensional analyses and reports with embedded business intelligence. Customized analytic components monitor existing business processes, as specified by the business user. The Adaptive Action Manager invokes selected real-time actions, or sets of actions, based on predefined rules and policies.

IBM Tivoli Business Systems Manager

http://www-306.ibm.com/software/tivoli/products/bus-sys-mgr/

This product provides a set of specialized dashboards that are able to monitor the health of the most critical IT services and any associated service level agreements.

In particular, you can:

- Monitor and programmatically discover and maintain resources and relationships within an IT infrastructure
- Leverage data from existing Tivoli and third-party monitoring products
- Display instantaneous (real-time) service level status and root cause analysis
- Provide end-to-end systems management support (distributed to mainframe)
- Utilize a Web console to manage IT from anywhere at anytime
- Display resource relationships in tables, hierarchical trees, hyperviews, and topology views
- Auto-discover new resources and auto-populate business systems

IBM Tivoli Service Level Advisor

`http://www-306.ibm.com/software/tivoli/products/service-level-advisor/`

IBM Tivoli Service Level Advisor is designed to provide predictive service level management capabilities by enabling you to proactively predict when SLA violations are likely to occur and then take corrective actions to avoid an SLA violation. Product features enable you to:

- Define service level agreements easily using an SLA wizard
- Integrate service level data with availability data on the IBM Tivoli Business Systems Manager executive dashboard to increase executive knowledge of IT as it relates to business objectives
- Provide SLA evaluations as often as hourly
- Take a proactive approach to service level management by utilizing a patent-pending trend analysis algorithm
- Provide enablement of mainframe and multivendor distributed systems management data for true end-to-end service level management

IBM WebSphere Portal

`http://www-306.ibm.com/software/genservers/portal/extend/`

The IBM WebSphere Portal is a comprehensive suite of components that helps you to quickly build scalable portals. You can create multiple portal sites on one instance of WebSphere Portal. Each site has its own URL, look and feel, pages, users and groups, and search index. All sites can share the same software and hardware, which lowers capital, maintenance, and administrative costs while expanding the business value of portal to new communities. These products provide modalities to combine people and applications at a process level. The Portal navigation paradigm is not only role-based but also includes workflow orchestration that presents users with the tasks they need to complete and all information and applications necessary to complete the task or decision quickly.

IBM DB2 Alphablox

http://www-306.ibm.com/software/data/db2/alphablox/

This product adds a number of capabilities to the IBM business intelligence portfolio. It adds a set of components, based on open standards, that allows you to deliver integrated analytics. It also enables you to broaden and deepen business performance management capabilities across organizations, as well as provides dynamic insight into your respective business environment. DB2 Alphablox provides various Blox components, which are modular, reusable components, as well as an application framework, a powerful programming model, and a variety of development tools for assembling analytic applications.

IBM WebSphere Information Integrator

http://www-306.ibm.com/software/sw-atoz/indexW.html

This is a comprehensive platform that includes all required capabilities for Information consolidation, data event capture, search, and analysis. Data consolidation is supported in several forms. In particular, the platform provides:

- Federated access to multiple disparate content management and workflow systems.
- A complete platform for deploying repository-spanning applications and workflows.
- Bidirectional access to content and workflow, as well as the underlying functionality.
- Pre-built integrations to more than 20 content management and workflow systems.
- A rich set of functions spanning multiple repositories, including federated search.

ETL-Replication provides two solutions to replicate data from and to relational databases:

- SQL replication, where committed source changes are staged in relational tables before being replicated to target systems
- Queue replication, where committed source changes are written in messages that are transported through WebSphere MQ message queues to target systems

In the federation integration modality, the federated data server features include powerful cost-based query optimization and integrated caching.

Search and analysis components provide enterprise search middleware for powering intranets, extranets, and corporate public Web sites, and a rich platform for building high value text analytic solutions.

Finally, this platform provides the ability to capture database changes in DB2 UDB by reading the recovery log. It formats the changes into XML messages and publishes them to WebSphere MQ. Any application or service that integrates with WebSphere MQ directly, or supports Java Message Service (JMS), can asynchronously receive the data changes as they occur.

IBM DB2 Data Warehouse Edition

http://www-306.ibm.com/software/data/db2/udb/dwe/

This is a comprehensive platform with all the functionality required to build a business intelligence infrastructure for developing data warehouse-based analytics and Web-based applications with embedded data mining and multidimensional OLAP. Some of the features included in this platform are:

- DB2 Alphablox rapid assembly and broad deployment of integrated analytics
- DB2 Universal Database Enterprise Server Edition
- DB2 Universal Database, Database Partitioning Feature (large clustered server support)
- DWE OLAP (OLAP acceleration)
- DWE Mining (powerful data mining)
- DB2 Query Patroller (rule-based predictive query monitoring and control)
- SQL Warehousing Tool for visual design of intra-warehouse, table-to-table data flows, and control flows using generated SQL
- WebSphere Information Integrator Standard Edition to provide native connectors for accessing data from heterogeneous databases

2.4 Performance Insight

Through customer engagements, market research, and analyst feedback, IBM has identified four common and basic challenges that companies can address with innovation and optimization right now and continue to build on to realize additional value in the future.

Each challenge, at the same time, constitutes a possible starting point in introducing IBM Business Innovation and Optimization. These are known as BIO on-ramps.

- **Process Innovation** to deliver continuous innovation and improvement
- **Performance Insight** for making more effective decisions
- **Business Alignment** of business objectives
- **Operational Management** to manage business operation disruptions effectively

For the scope of this redbook, we have chosen to focus on Performance Insight.

One of the challenges, or business problems, that performance insight attacks is concerned with the ability to optimize decision making. Having acquired and evaluated, in real time, data flowing from process operations, we can make faster and more informed decisions. Think of performance insight as the right balance and combination of business process management and business intelligence.

The value of the IBM Performance Insight itself remains in having the correct tools and an approach to integrate business process management and business intelligence to have a fully dynamic and integrated operational and strategic action loop. Having access to real-time data that is actionable and the ability to perform *inline analytics* is a distinguishing capability of performance insight.

Consider the example of a company looking to provide more efficiency in its Product Returns process, and align cost reduction, revenue, and service level goals. Studies suggest that, in some industries, the percent of returned products with respect to the total sales volume can be high, perhaps up to 20%. Returns occur for a variety of reasons, including clients simply changing their minds, errors, product damage during shipment, wrong quantity, and so forth. The Product Returns process could be expensive for a company, and there are many factors that contribute to that expense. Just to give you an example, in some industries, companies declare that shipping costs of the products returned back from customers is 4-5% of the total shipping costs paid by the company.

To effectively respond, notifications and intelligent support could be used. This business intelligence adds business context to the business process. That could, for example, drive action to the Call Center representative to suggest a solution such as product replacement, rather than a refund. Or it could drive action to decide to change the supplier or shipping operator.

Performance insight is at the center of a business performance optimization task. Performance insight aims to combine information coming from business processes and business applications and, by using business intelligence and data mining capabilities, generate insights to optimize business results. The novelty concerning this approach, with respect to the current state-of-the-art approach in the industry, is the fact that this is not a simple combination of business process management and BI, where both sides can supply services to the other. Rather, it is the integration of the two.

The following chapters continue the investigation into performance insight by providing additional details of the integration of BI and business process management. Then, that insight is put into action with a case study for an example of how it might be used in practice. The case study describes and demonstrates how you can effectively implement performance insight using IBM products and technologies.

3

Performance Insight

In this chapter, we introduce Performance Insight, one of the on-ramps to Business Innovation and Optimization. From a simplified perspective, performance insight is the integration of Business Process Management and Business Intelligence (BI). We describe the different elements of Performance Insight, and how it enables faster and improved decision making by providing the right information to the right people at the right time. It is this faster and improved decision making, and new business insight, that enables organizations to be more proactive in their business process management and gives them an improved opportunity to meet their business performance measurements and enterprise business goals.

3.1 Getting the information for performance insight

Most decisions made in corporations today are an integral part of some business process. And those decisions today are not just reserved for management, there is more and more reliance on the front-line workers to take appropriate action. However, this has generated a more significant demand for the availability of timely and relevant information, in the right context and from various sources, to enable both management and workers to make informed and effective decisions.

In addition, technology and leading-edge organizations are enabling decision making to be taken a step further. And that step is the enabling of decision making by the business processes themselves. That is, with improved business processes and their integration with IT, many decisions can now be made inline, in real time, as the process executes. This is a significant advancement in efficiency and effectiveness of the business processes.

As this technology advances and improves, these business processes continue to improve. Many times today, the data that is provided by the application modeling the business process does not supply all of the information and contextual insight required for these inline decisions. One way to satisfy this requirement is by providing an analytical service.

Included with such a service are dashboards that can provide an aggregated real-time view of the performance of the organization and indicate actions that can be taken to better manage to successful achievement of the business performance measurements.

As an example, with WebSphere Process Integration we have the ability to model, execute, and monitor a process, and a means to optimization. Performance insight takes this to the next level by introducing business intelligence at the point of decision. Then, based on the availability of real-time data, alerting, inline analytics, and more informed decision making, organizations can more effectively optimize the business.

3.1.1 Roles

When discussing how business intelligence can be used within a business process, we need to look at different roles of users. In this section, we discuss two primary roles in a performance insight solution.

Direct participants in the business process

Who are the participants in a business process? These are the front-line workers who execute the activities defined by the process. They need to leverage business intelligence at the point of decision. And, it is critical that business intelligence is provided in context with the business process and the user. Most users probably do not even notice that it is business intelligence, but rather see it simply as relevant information delivered to make a business decision.

For example, assume that a customer call agent needs to make the decision regarding a customer request to return a product. It is not only important for the customer agent to have all the information about the product and the customer from the CRM system, but it is also relevant to see historical purchasing information about that customer. This is to help determine the relative value of that customer, from a purchasing perspective. This type of information should be contained in a data warehouse and is extremely relevant to making a decision concerning the return action and the possibility of granting a return exception. That is, even though the product warranty period might be past or the customer has no receipt, or the date is past the returns policy, we might still decide to let the customer return the product.

Business Process Monitoring

Who monitors business processes? These are company managers and executives who are not directly involved in the business processes but want to monitor the performance or the results it yields. Therefore, a critical element of Performance Insight is comprised of the dashboards that provide role-based views of the current state of the business.

However, more is required. You need to add the element of measures that describe individual or aggregated business processes as well as metrics. These come from an enterprise data warehouse or external source, in the form of BI. It is the combination of this BI from the enterprise data warehouse with the state data from business monitoring that truly can give an enterprise view and the desired performance insight.

With this insight, management has the information required for effective decision making. They can now be proactive in leading the company to success in achieving the business measurements. This is what the company needs, and the stakeholders are demanding.

3.2 Performance insight components

In this section, we discuss the relevant components of performance insight, which are business intelligence and business process management.

3.2.1 Business Intelligence

In the business world today, many of the decision makers need to have access to accurate and timely information in order to achieve their business goals. And, in many organizations, more and more of the decisions are being pushed down to the front-line workers. Enabling those front-line workers with more easy to access, understand, and use business intelligence is essential for effective decision making and business success.

Historically, business intelligence has primarily been used by analysts and management, who have analyzed data through the use of complex tools and spreadsheet applications. But now with decision making involving a larger range of employees and business roles, the major business intelligence vendors have focused on providing full function business intelligence suites. These enable the decision makers to access source data from almost any environment and use reporting, analysis, OLAP, and data mining through a single, easy to use, integrated solution package.

Also, many of the business intelligence suites have moved to Web interfaces to enable an even broader range of users to access data. The primary focus now is on providing easier to use interfaces to further enable and broaden the range of users for data access, analysis, and business intelligence. It is essential to have customizable user interfaces and integrate business intelligence tools into existing dashboards and enterprise portals to provide users with a single view of the enterprise.

It has also become increasingly important to personalize the business intelligence interfaces to users and user groups to provide them the data that is specific, relevant for their job, and supported by the functionality that is necessary to properly analyze the data for improved business decision making.

Providing appropriate right-time analytics to give users accurate and up-to-date data has also become critical. Today, data warehouses are able to handle much larger volumes of data and perform aggregations in near real-time. Through information integration capabilities, users can also now access both structured and unstructured data through a common interface.

3.2.2 Business Process Management

Business process management, inside the Performance Insight on-ramp, has resulted in enabling very important and strategic capabilities. This is because the organization can implement their business processes company-wide and end-to-end. And, the organization can monitor the processes to enable closed-loop feedback for ongoing improvements. This is the required flexibility we have discussed that becomes even more powerful when integrated with IT support.

What does this mean? Well, until now, the prevailing point of view has been that business strategy and needs of the business lead, and technology lags in providing support to achieve innovation and competitive advantage. This has been the conventional wisdom for quite some time. However, there are new external pressures caused by emerging technologies, global sourcing, mergers and acquisitions, and the requirement for increased regulatory compliance.

Organizations are coming to realize that by integrating their business strategy, design, processes, and technology, resources used to support one part of the business can be shared, or leveraged to support other parts of the business. They can then achieve even greater business performance, not only internally, but beyond the four walls of a company with their suppliers and other business partners.

With business process management, the organization needs to think of their processes in terms of cross functional areas. This enables them to integrate the business processes throughout all of the organization and allows them to reflect company strategical objectives inside the business processes. With the IBM business process management approach, your organization can gain the benefits of market-leading software, industry expertise, and best practices at each of the following key stages of the life cycle:

- Governance and process. Improvement starts with the identification of business goals. This enables an understanding of the reasons for the very specific definitions, capabilities, and modeling of the business processes.
- Model. By using IBM software to model the business processes, you can simulate a process to understand how it works prior to implementation. This enables you to define the appropriate key performance indicators (KPIs) to monitor and manage the efficiency, effectiveness, and success of the processes.
- Assemble. After the process has been optimized, you can reuse IBM software to assemble the IT assets necessary to support the process throughout your organization.
- Deploy. Next, you deploy the process, policies, and resources that you have modeled and assembled. These can include data models or models for strategic business operations.
- Manage. After deploying a new process, you embark on the final and probably most important stage - management. Dashboards, KPIs, and alerts tied to real-time event-based data can enable your users to monitor and manage process execution and performance. You can analyze the progress with the process and use this information for continuous improvement. The

IBM software can even offer suggestions for actions to take based on the status of the KPIs being monitored.

With the business process management approach, you can enable integration of people, process, and information, and align IT process, tasks, and activities with strategic goals and initiatives. With business process management, you can deliver real-time, contextual information to decision makers, while developing the flexibility to quickly and effectively respond to changes in the market, customer demand, and the strategies of your competition. And, you can gain deep process and operational insight to enable risk management and change.

3.3 BI and Business Process Management

Performance insight is about the integration of real-time process performance data with business contextual information as the means of optimizing decision making so organizations can more proactive, taking action faster to avoid or minimize problems rather than reactively trying to minimize the impact of those problems.

In this section, we discuss the integration of business intelligence and business process management. There are levels of integration. For example, there is integration at the data level and integration at the visual level. We discuss both of these areas as the means to performance insight.

Integration at the data level is critical. This integration provides the overall enterprise view that enables the required decision making support so that appropriate action can be taken at the appropriate time. It is the integration of the process data with the enterprise business intelligence.

This enables a key decision making capability, called *inline analytics*. With this capability, you can begin to automate more of the decision making. The integration of the BI from the enterprise data warehouse with the real-time process data makes available the critical information for these automated decisions. Of course, there is more than data required for this, but it is a key support resource.

We also discuss visual integration, in particular, the use of dashboards as examples of technologies that can help achieve a level of integration. Here the data comes together from various sources and is displayed on a portal. The action resulting from that information can either be automatic or manual, depending on your particular capabilities.

3.3.1 Data integration

Most companies have done relatively little integration between their business processes and business intelligence. This is primarily because the enterprise process applications have traditionally been delivered by ERP vendors and middleware companies, and business intelligence has been delivered by specialized BI vendors. So, there has been less opportunity for the possibility of such integration. In addition, the data management and business process development activities have been in completely different departments or areas of the organization, limiting the interaction and incentive for integration.

Technology to the rescue! This integration is becoming easier with such initiatives as service oriented architecture. Integration can now be accomplished even with loosely coupled application environments. SOA has enabled the creation of easy to use process and business intelligence services that can be used without having to fully understand the underlying implementation details or having detailed domain knowledge.

Note that we are not saying that integration is now easy. We did say it is becoming easier. But there are still challenges, for example, the integration of process data with business intelligence data. This is because correlating data from a real-time process with data that has gone through several transformations prior to being stored in a data warehouse can present a challenge.

One example of such a challenge is when product or customer numbers differ. But, there is help here as well. Master Data Management (MDM) has made this significantly easier because it enables you to share a single master data set across business processes and the data warehouse. This is another reason you should be actively implementing MDM, if you are not already doing so.

We also have better tools for accessing and integrating data from multiple heterogeneous data environments and for moving closer to a real-time data environment. For example, you can accomplish this through the use of tools such as WebSphere Message Broker (Enterprise Service Bus) for providing a common structure for data delivery, or, with WebSphere Information Integration. This enables you to access heterogeneous data sources as though they were DB2. Using WebSphere MQ for enabling a continuous flow of data into the enterprise data warehouse can also help you move toward real time.

Another strategy is the expanded use of an operational data store (ODS), which can be used to keep more real-time operational data, which is then joined with data from the enterprise data warehouse.

There are many techniques and technologies available from IBM to help you in these integration activities. The time to move is now.

3.3.2 Inline analytics

Spreadsheets are one of the most used tools for performing analytics in business today. They enable the easy development of simple, but very powerful analytics such as formulas and macros. A major problem with this approach is that it requires specific skills and capabilities, not only to use the spreadsheet applications but also to have a good and accurate understanding of the data. The data is used offline and outside of the business processes.

This means that spreadsheets are basically independent silos of data, maintained by individuals working independently. They are not monitored or managed on an enterprise basis for accuracy, consistency in data types and definitions, or currency (orchestrated date and time of update). Their use across the enterprise is limited, because they are not developed or maintained with a common or standardized approach. Therefore, accuracy, credibility, consistency, currency, and definitions of the data should all be called into question.

Today, business intelligence applications are used in a relatively small percentage of corporate environments. This is particularly true when considering the front-line workers who are executing the business processes. One of the primary reasons is that business intelligence applications might be more difficult to use because they need data at the right time and in the right context to enable them to make decisions, and many times there are also complex analytics involved. Therefore it is critical that businesses embed business intelligence tightly into the business processes - in the form of inline analytics and analytic applications.

Inline analytics represent the ability to provide analytical services at the point of decision, to optimize decision making. This means that analytics need to be an integral part of the business process. As a user or system is at a point of decision within the business process, analytics are delivered in the appropriate business context.

Types of inline analytics

In this section, we discuss three ways to deliver inline analytics:

- **System-driven analytics** refer to a system that programmatically consumes an analytical service for decision making. That is, decisions are made using information as a service. For example, consider a customer who would like to place an order. During the process, the system programmatically suggests payment methods, using a data mining solution, and based on a score from the customer payment history.
- **User-based real-time analytics** are analytics provided within a process step that requires user intervention. It is critical that the decision is made in real time. For example, a customer calls into the Call Center and requests an

exception on the return policy. The Call Center agent is provided with real-time information about the customer's historical purchasing information to help the Call Center agent determine if the exception should be granted.

- **User-based complex analytics and guided analysis** can be time-consuming. For example, they often require use of OLAP to analyze the problem across multiple dimensions or for analysis across multiple data sources as they search for appropriate data.

  For example, assume there are an unusually high number of returns for a particular product. To find the root cause, the analyst must analyze the problem across different dimensions. A skilled and experienced analyst can do this. But, to enable other lesser skilled workers to perform these analyses requires some help or additional tools.

  One type of help, or tool, is called *guided analysis*. In simple terms, this is an application that has embedded in it the analysis techniques of a skilled analyst. It can provide suggestions and analysis paths to help guide the lesser skilled worker to the root-cause problem. For example, it can provide information about any similar past problems and suggest appropriate data sources to analyze and specifics on what they should be looking for.

  Guided analysis applications can be very valuable in enabling a wider range of workers, rather than only more highly paid analysts, to perform problem analysis. Problems can be resolved more quickly and for much less cost.

3.3.3 Dashboards

We have discussed data integration, but there is also visual integration, that is, integration of information by displaying it for a user. Perhaps there are multiple displays, possibly in a portal, that the user then integrates visually.

Although this type of integration can be done via dashboards, that is not the primary goal of dashboards. Dashboards are the desired output mechanism for the integrated data that we have just previously discussed. The message here is that dashboards can play multiple roles.

Dashboards are intuitive, informative, and possibly interactive displays. For an example, refer to Figure 2-7 on page 33. They include gauges and meters that enable the visual display of information for fast and easy understanding. For example, this is where alerts can appear to enable fast action to be taken. Visually noting that a performance indicator is approaching a threshold value can enable action to be taken before a problem occurs or becomes critical. This is proactive management. It can help eliminate the more typical reactions to a problem that has already happened, in an effort to minimize the impact.

Enterprise portals, such as WebSphere Portal, provide a set of standards-based interfaces that allow other application and data to integrate into the portal. JSR168 is the standard today that allows any application to create portlets and access users, configuration, portal, and portlet settings through a standard application program interface (API). There are certainly different levels of integration. Almost any application can be viewed through a portlet, but, for a performance insight solution, a higher level of integration is required, because the process flow has to be seamless between business process management and business intelligence.

To achieve such integration, the following are considerations for the business intelligence and business process management components:

- Portlet to portlet communications.
- Get user portlet configuration properties and personalization parameters from the portal and portlet.
- Call portlet APIs from the application. For example, to enable right-click menus or click events.
- Decompose the application into loosely coupled portlets.
- Provide a common look and feel across applications.

DB2 Alphablox and WebSphere Process Server are built on an open architecture that allows embedding into enterprise portals.

4

Business Process Management

It is a dynamic global marketplace, and business trends today demand a continuing evolution of business processes and their management. In addition, business cycle slowdowns require companies to examine and streamline their business processes to minimize costs to stay competitive. And, these processes more and more need to be updated and automated for improved efficiency and effectiveness. This cannot be accomplished with processes that are not properly documented or are hard-coded in a particularly technology.

In the IBM Business Consulting Services publication, “The Agile CFO”, 889 CFOs and senior finance professionals from 74 countries were surveyed. It revealed that over 60% of participants indicated they have yet to implement enterprise-wide standard policies and rules or extend common processes across the entire enterprise. Additionally, more than 80% have not pursued enterprise-wide process simplification or expanded use of functional best practices across the enterprise. Over 70% of participants have not yet reduced the number of common platforms, rationalized budgeting and forecasting tools or reduced the number of enterprise resource planning (ERP) systems enterprise-wide.

Naturally, this fragmentation and lack of standardization results in various *versions of the truth*, manual data reconciliations, and ineffective use of technology, inhibiting the ability of Finance to influence decisions and deliver insight. For access to this publication, see “Other publications” on page 427.

In this chapter, we discuss how the business of the organization can be realized through processes. And modeling those processes allows them to capture how they run their business more formally. It is then that we can begin to understand what a process is and the elements which comprise it. The information helps provide a conceptual framework for how organizations can describe and define their businesses to be focused on processes. Along with that, we present some of the advantages and define solution content for business process management.

4.1 Defining a process

We start with the basics to enable a common understanding of the material. This is very basic, so bear with us. However, we wanted to present it for those less familiar with business processes and associated terminology. So, first we describe a process.

In simple terms, you can define a business process as a set of defined activities that a business unit performs in response to an event. Within the business process, there is a logical set-of-work performed at a particular point in time. The process also describes how to perform those work activities. For example, it specifies how a business leverages the capability of its active resources (people, knowledge, and application systems) and passive resources (equipment, physical assets, and capital). The overall objective is to realize strategic capabilities, support value propositions, and create a valuable outcome.

Elements of a process

The following are some of the elements of a business process:

- **Input:** The material or information required to complete the activities of the process necessary to produce a specific end result.
- **Output:** All the data, information, and physical assets that the process generates. This output represents value for the organization, and contributes to the attainment of the business measurements and goals. It also represents events and actions, or the results of those actions.
- **Events:** These are notifications of some occurrence of importance. For example, an indication. They can occur before, during, and after the execution of a process. They might indicate the start, intermediate status, or end of a process activity. An event could be an action resulting from the completion of another process (or process activity), the meeting of a certain condition, or the arrival of a particular point in time.
- **Sub-Process:** A defined process, or process step, inside another process. A sub-process is defined when it is not possible to represent the scope of work with only a set of activities. The sub-process has the same elements as the process.
- **Activity:** The lowest level of work in a process.
- **Resource:** Represents the person, organization, equipment, or system performing the work in a process.
- **Performance Metrics:** Attributes that help and guide the process owners in controlling the process and determining if the process is efficient and effective. That is, determining if the process meets the stated performance

measurements and business goals. The purpose of the performance measurement is to:

– Determine that the actual input to, performance of, and outcome from a process is as planned.
– Understand how well the process is meeting customer and stakeholder expectations of performance goals.
– Identify potential areas of improvement in the process.

These elements and their interactivity are depicted in Figure 4-1.

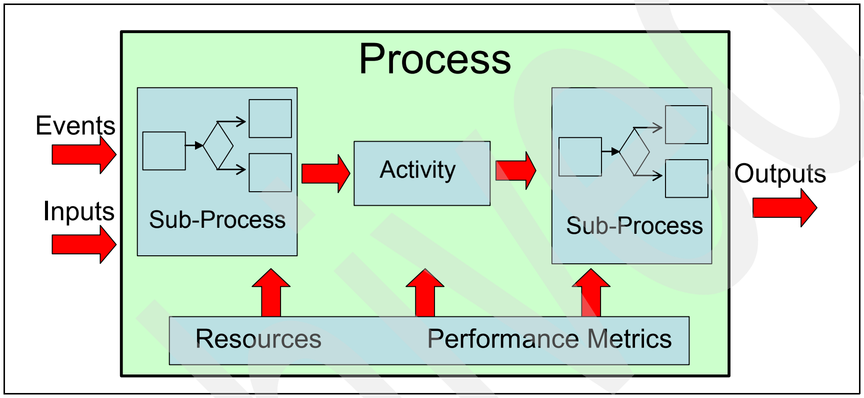


Figure 4-1 Process elements

These elements are defined when you create a process or document your current process. To help you understand more about these elements, we provide a process example.

The example is about a company that sells electronic products. This specific process describes the activities that occur when a customer, who purchased a product, wants to return the product because of a problem. In this instance, the company has the option to either repair the product or exchange it for a new one. The example product inspection process is described in Table 4-1.

Table 4-1 Returned product inspection

Process name	Returned Product Inspection.
Inputs	Returned Products and customer information.
Outputs	Determination that either a new or repaired product should be given to the customer.
Event	Product enters the inspection activity.
Process descriptions	When the product arrives, the inspector inspects the product per the appropriate documented procedure. This requires a product identification number. Following the inspection, the product is either repaired and returned to the customer, or scrapped. Other activities in the process determine whether a new exchange product or a refund is sent to the customer.
Resources	Inspector and inspection document.
Performance metrics	Time and cost to inspect the product and either repair the product or scrap the product.

The real-time business process activities generate operational data. That data is used to understand the results, in terms of time and cost of the activity. That data is also used to generate a historical data source for potential subsequent analysis. This data can be stored in a real-time operational data store during the life of the process and then transferred to the enterprise data warehouse as history data.

In the next section, we explain how business process management uses historical and real-time data to improve the business processes. But also how it is used to monitor and proactively manage the business processes to achieve the business measurements and business goals. It is only through the use and proactive monitoring and management of defined business processes that organizations can be confident about the attainment of these goals.

4.2 Managing the business processes

Business Process Management enables an enterprise to be flexible and responsive to ever-changing On Demand Business through the optimization and automation of business processes to:

- Identify and eliminate redundancies and bottlenecks
- Decouple business integration logic from the implementation code
- Increase portability and decrease costs by use of industry standards
- Minimize manual tasks

- Quickly implement new business rules and processes
- Monitor and manage process performance, using KPIs and alerts

When describing business process management, there are two primary perspectives. One is relative to the *Management Disciplines* and the other is relevant to the *Technology Platform*.

Management Disciplines: Business process management is a major initiative in industry today and is seen as a valuable approach for gaining better insight about, and control over, business operations. Significant effort by management is expended in developing business strategies and goals and distributing them throughout the company. The problems have historically come when trying to monitor and manage the execution of those strategies. This is because many organizations do not have the processes or management tools in place to accomplish it.

Another issue is that planning and budgeting cycles are not flexible or fast enough to satisfy the fast changing business requirements. Plans and budgets are many times out of date before they are completed. This can happen for many reasons, one of which might be that inappropriate and non-integrated tools and methods have not kept up with current practices. There are many companies, for example, that base their complete planning process on a series of spreadsheets linked together over different computers, and even departments. What is required are specialized software solutions developed specifically to define, develop, monitor, and manage business processes.

Business process management replaces traditional views of business based on organizations conceived of functional and departmental areas, with their metrics and procedures based on cross-functional core processes aligned with high level business objectives and enterprise strategy. This is because traditional views can present problems when they need to expand the business process across organizations within the enterprise. It is not possible to have global visibility of the enterprise when processes and measurements have only an organizational focus. Figure 4-2 on page 63 shows the traditional view conceived with only functional areas, such as Financial, Technology, and Services, with a process view that crosses all the organizations.

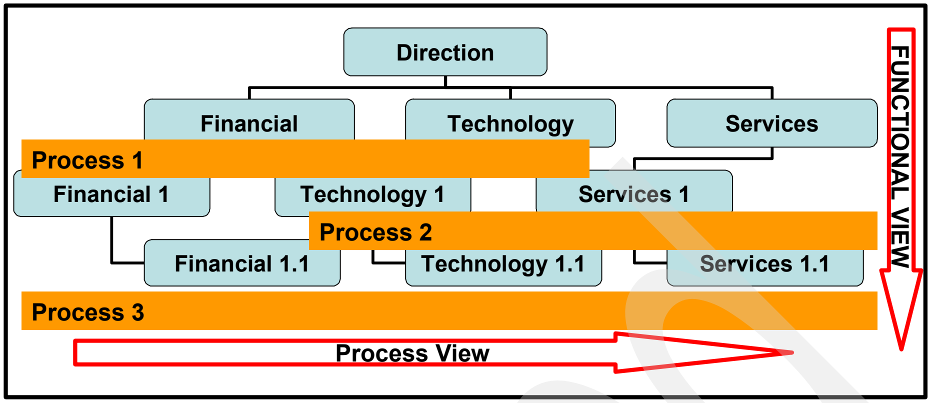


Figure 4-2 Functional and process view

Technology platform: This approach provides a convergence through integration and enhanced technology to help streamline the business transformation. It provides a set of software tools needed to optimize performance, make abstract performance goals more concrete, connect them to process data, automate and monitor process activities, and provide a platform for agile performance improvement. This approach also perceives IT as a facilitator of the business.

With business process management, the organizations transfer business strategies to the business processes so that each component is involved in the fulfillment of the corporative objectives. The components that can be used are suppliers, clients, technology, and workers. This implies that those involved in decision making in the organization obtain information in time and formats to enable them to determine the best direction for the business. These organizations can model and analyze the end-to-end process as whole. They provide modeling tools that allow business analysts to document and define measures for the existing, and proposed new, processes. Business process management is all about making the processes that are core to your business work better.

Business process management also combines business processes, information, and IT resources, aligning your organization's core assets of people, information, technology, and processes, to create a simple integrated view. This includes the real-time intelligence of both its business measurements and IT system performance. This integration of resources allows your organization to obtain business information faster, respond more quickly to market trends and competitive threats, and improve operational efficiencies for better business results. Business process management enables your organization to operate more *effectively* and *efficiently.*

4.2.1 Benefits of business process management

Business process management is a management discipline for managing business processes. It can enable a company to become more agile and better integrated to see the business measures and goals as they relate to the entire enterprise. Being agile can enable an enterprise to bring new products and services to market quickly, to respond rapidly to changing demand, and to be proactive rather than reactive. This brings with it many benefits, such as:

- Allowing you to extend the scope of process automation and management across the IT barriers that historically have separated departments. That is, business and IT can be more integrated. The gap between business and IT is diminishing because both areas can now work on a common business model.
- Making process performance visible at the process level, tracking process data, and monitoring it relative to selected key performance indicators, and aggregating it for better management on, for example, graphical dashboards. This visibility can be segmented into different views for different functions, such as for process owners, system administrators, and business executives. This enables a better understanding, because each functional area has different business process requirements.
- Providing the capability to identify and eliminate data redundancy and bottlenecks, because it is possible to identify them during development of the process rather than after they are operational. That is, you can simulate the processes before releasing them to operations.
- Reducing risk by gaining an understanding of process impacts prior to operationalizing. The entire organization agrees about the process because multiple people on multiple teams around the company can view and contribute to the development of the business processes.
- Visualizing actual process performance against key performance indicators. You understand the process status because you see what happens in real time, so you can make decisions more quickly and more accurately.

Business process management, from an IT platform perspective, does not replace your existing IT investment. Rather, it enables you to orchestrate process actions to make end-to-end processes more efficient, more flexible and agile, and more standardized and compliant. It is *efficient* because it automates manual tasks and makes sure the most important tasks are done first and on time. It is *agile* because executable process models are not built with complex code but composed graphically, similar to a flowchart, so they can be quickly and easily changed. A key enabler of business process management is service oriented architecture (SOA). We discuss more about business process management and SOA in the following sections of this chapter. It is *compliant* because the process logic is based on business rules reflecting policies and best practices.

Other IT benefits of business process management are:

- Business integration logic is decoupled from the underlying implementation code. Creating process independence helps facilitate the best alignment between business process modeling and actual enterprise implementation. New and changed processes that are modeled can be more rapidly implemented in the enterprise infrastructure.
- Increased portability and decreased maintenance cost because processes are based on industry standards.
- Process implementations are automated resulting in the elimination of manual deployment tasks. Key in automating processes is to focus on reusable processes, or process elements. This not only makes implementation easier, faster, and less costly, it creates efficiencies in maintenance and upgrades.

Business process management enables the collection of information about the process executions. For example, significant events that occur during process executions, such as the start and completion times of activities and the input and output of the activities, are logged. This information represents the knowledge of the organization through time. In addition, this information is stored so it can be reused. It can then become input for the resolution of problems or in capturing opportunities for future improvements. The information generated by the business processes enables the organization to discover problems and understand them. Upon their resolution, the information about the problem environment can be used as a way to predict potential problems in the future.

The information collected about business process executions can be useful in establishing an approach for the creation of a business process-oriented data warehouse. In the next section, we discuss how the addition of business intelligence can help to optimize the business process.

4.2.2 Business intelligence and business process management

Historically, companies have primarily resorted to the use of operational data to execute and manage business processes. Today, companies derive significant value by using pertinent data from anywhere in the enterprise. This is a primary benefit of developing an enterprise-wide business intelligence initiative.

So now you can combine business process management and business intelligence to optimize your business processes. However, this does not simply mean bringing together information from various areas of the enterprise. It means integration.

When we say integration, we mean that business intelligence can be integrated into, or included as a part of, business process management. One example of this is the use of inline analytics, which can analyze the process as it executes. This is what can enable problem prediction, giving insight, for example, to the potential of missing a performance target. Further information can then help to identify and implement a solution, even before it becomes a real problem. We refer to this as gaining *performance insight*.

Gaining performance insight

Many companies struggle to use information effectively. Although they might have automated many of their individual business processes, for example, to improve their supply chains, reduce product cycle times, better understand their customers, or lower transaction costs, this has often left redundant and fragmented information across their enterprise. These enterprises have their business processes running on multiple servers, applications, middleware, databases, and operating systems. And although they can communicate with one another, it is often very difficult to get a comprehensive and unified view of the enterprise.

Implementing a data warehouse and using business intelligence and data mining technology can provide a significant benefit. For example, use it to:

- Analyze the performance and quality of the resource, for example, by comparing the process activity duration times across different resources.
- Understand and predict exceptions. BI can be used to understand the real cause of problems, and, hopefully, avoid them based on knowledge gained from past process behavior.
- Optimize processes. With BI, you can discover conditions under which specific paths or sub-paths of the process are executed, so you can redefine the process.
- Improve process execution times. Analyze process execution times and quality testing configurations of the system, assignment of resources, and dynamic adaptation of the process.

Most of the data in an organization has originated from the operations of their business processes through time. This data, in many cases, is stored on different platforms and based on different technologies. It is in these types of environments that the heterogeneous information integration capabilities of BI work to enable a *single view* of the business processes. With this view, and the real-time information available from the processes, managers can now effectively manage. KPIs can be monitored, alerts can be generated, and action can be taken to proactively keep the enterprise on target to meet the business measurements and goals.

It is this information, process flexibility, and proactive management that can deliver success. This is depicted in Figure 4-3 on page 67. This is not to imply that action can only be taken after insight. It can be taken anywhere along the spectrum. It is simply that at that point the action is more insightful and therefore has more business value.

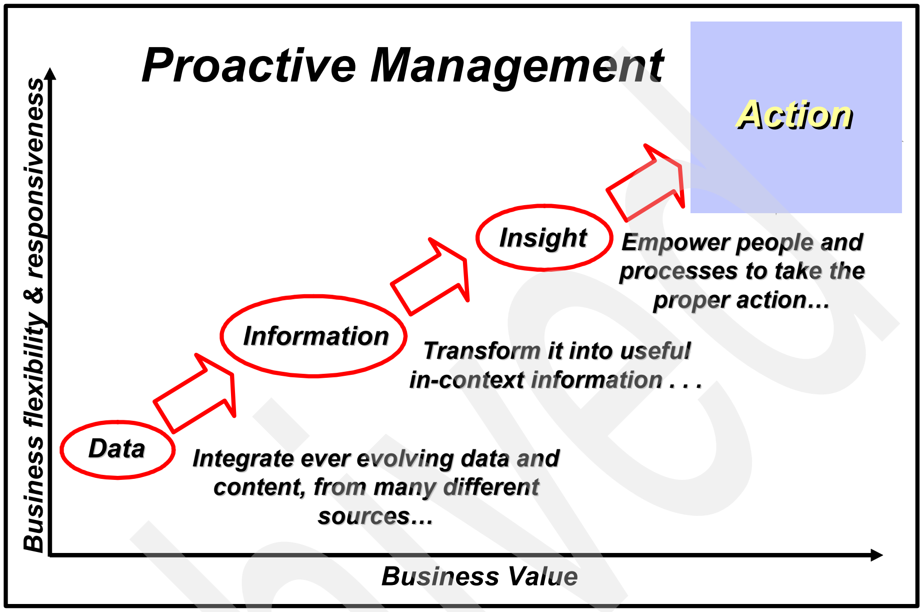


Figure 4-3 Using insight to increase value

The business world is moving very fast, requiring enterprises to have available all their sources of information to make the most informed decisions possible. To aid in this effort, IBM has established the *IBM Information On Demand Center of Excellence*. Using the Center's Information On Demand maturity model, organizations can determine where they are now - and where they must go to remain competitive. The model describes five stages of maturity of information use:

- To run the business
- To manage the business
- As a strategic asset
- To enable innovation
- As a competitive differentiator

For further reading about the *IBM Information On Demand Center of Excellent,* see “Online resources” on page 427.

4.2.3 Business process management functionality

The IBM business process management solution component support centers around four major tasks, model, assemble, deploy, and manage. In this section, we discuss these tasks along with their functionality. That solution component is depicted in Figure 4-4.

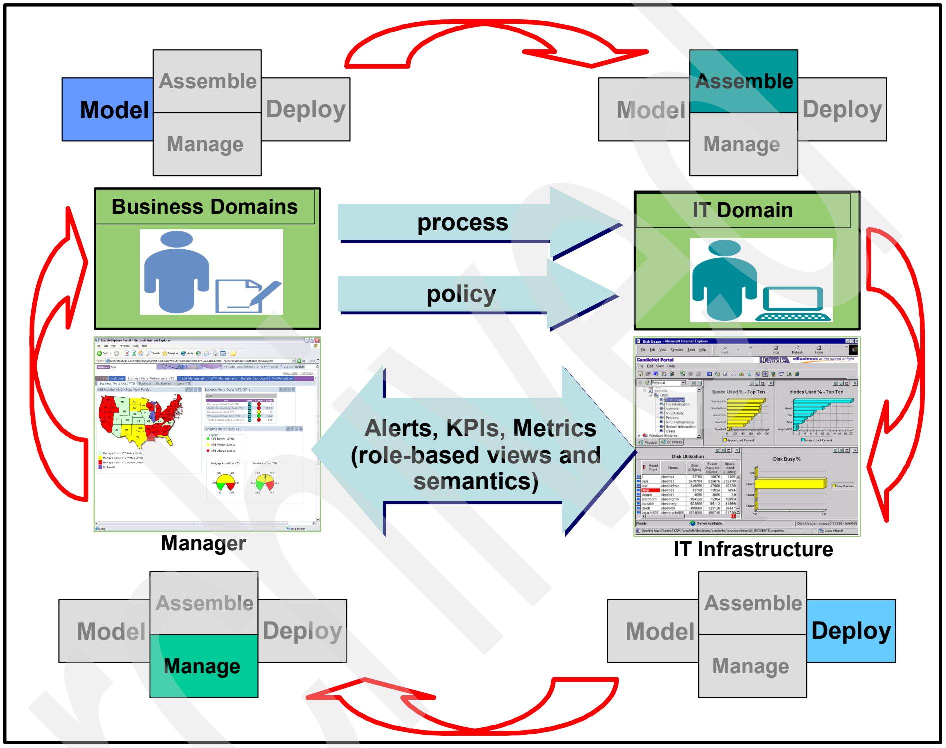


Figure 4-4 Business process management four major tasks

Model

A *model* is an abstraction of a real environment, or physical system, that has a specific purpose. It is a vehicle that you can use to capture a view of a physical business process and describe all the relevant aspects of that process.

It has characteristics such as:

- Purpose
- Perspective, or point of view
- Audience
- Content
- Level of detail
- Phase, as related to a life cycle

You can have a number of objectives when modeling a business process, such as:

- **Modeling for documentation and compliance**

  Here we document the process to better understand the business, to capture and document complex behavior, and gain domain expertise in the particular process. The output can be used for training, collaboration, and documentation.

- **Modeling for redesign and optimization**

  The objective here is to discover potential areas for process improvement and latent business value. The current state and proposed future state of a business process are documented, and they are compared to validate enhancements and ROI before committing resource. Measurable process metrics are established and tracked for performance optimization.

- **Modeling for execution**

  The business process should be changed to respond to business changes. After being changed, the business process is now ready to be passed to application, workflow, and business process development tools to be executed as a new running process. Linked real-time monitoring provides feedback on the process improvements.

Business modeling is a task that helps capture information that is valuable to the business. That includes such elements as business policies, key performance indicators, business events, and the action necessary to respond to those events and optimize performance.

There are many considerations when developing the activities that comprise the business processes that you are modeling. The following are examples:

- *Fully understand your current process*

  You should understand the current process and how it works. By clearly knowing what metrics are currently being used and how they satisfy the organization needs, you are able to better elaborate the changes required for the future business process plans.

- *Plan your process and model correctly*

  To plan for the process model, you must clearly understand the goals of the business and the organizations that comprise it. The following are examples of some of the questions that require answers:

  - What is the goal of the business model? For example, is the model to be used in order to identify areas of process improvements, or is it to be used as an informational tool to identify needs and in planning for future new processes.
  - What are the boundaries of the process model? Identify what the process model needs to include. Knowing the process boundaries enables you to more clearly define the activities that fall within those boundaries. This includes explicitly defining the specific inputs and outputs of the model.
  - What is the model point of view? That is, what is the exact use for the model? For example, is the model to be used for a business, system, or is it a particular business function? And, what is the level of detail does the model require? Is it high, medium, or low?
  - For whom is the process model intended? Identify the resources, or categories of resources, that use this process model. Be sure to capture all of the primary users, or the audience, such as business analysts, department and functional managers, and process owners.
  - What is the granularity of the process model? One way is to classify the granularity of the model as logical (identify the what) or physical (identify the how).

- *Understand the strategy to align the process capabilities*

  Obviously, one requirement is to develop a process that helps fulfill the business strategy and add significant value. How well you do this depends on how well you understand the strategy. The following is a list of some of the considerations, and how they relate:

  - *Strategic value propositions* describe the unique value that an enterprise offers to its customers, suppliers, and partners, that make the enterprise more competitive and successful in the marketplace.
  - Strategic capabilities define what the enterprise as a whole must be able to do and how well it must do it in order to successfully support the strategic value propositions.
  - The internal process capabilities describe what the various processes of the enterprise must be able to do in order to enable the strategic capabilities.
  - The resource capability describes which enterprise resources are needed to support the process capabilities.

– The business process design engagement team needs to design the business processes that support the process capabilities. This is done through an understanding and assessment of current processes and then the improvement or redesign of the current processes and the design of new processes. Business process design includes the specification of the capabilities of the business process enabler requirements (for an organization, knowledge and technology) that are necessary to enable the new business processes.

A summary of these considerations is depicted in Figure 4-5.

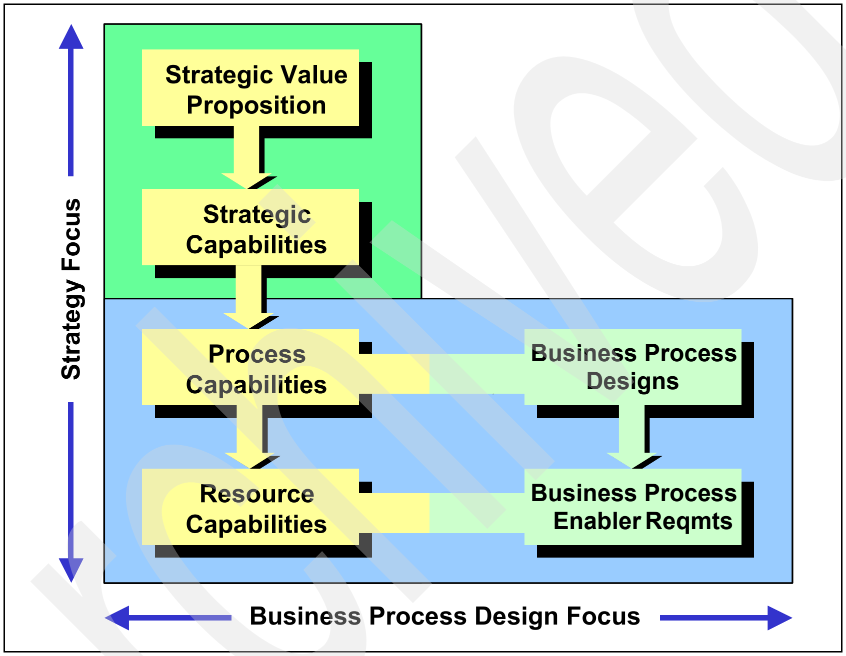


Figure 4-5 Strategic process capability

Assemble

After defining the processes and simulating them for optimization, you export them to the IT environment for the addition of technical information. At that point, you have integration between business and IT, with a common model. This common model helps to diminish any gap between these two areas.

In these steps with the IT community, the model does not change except that technical data is being added to generate a model implementation. The end result of the assemble step is an executable representation of the modeled process. Because the assembly started from a business-generated model, the implementation reflects essential characteristics specified by the business and supports capture and analysis of business-defined KPIs and other metrics critical to success.

Deploy

The deployment step results in the execution of the assembled IT solution on a process server. When a business process solution executes, the process server routes the service associated to each activity in the process model and tracks process performance every step of the way. Some steps in the process represent human interaction, and others represent automated functions of the business system. The server must choreograph the integrated human and programmatic activities.

For human interaction, the business process server automates and manages the workflow. For automated functions, the process server automates the business integration, mapping data between those systems involved.

Human workflow automation is a major contributor to business process management ROI. It accelerates cycle time, allows more work to be performed without increasing head count, and ensures that all processing conforms to established policies and business rules.

Business integration means that the diverse enterprise applications and information systems involved in the end-to-end process work in concert, even though they might not have originally been designed to do so.

Manage

The fourth task is management, which primarily refers to monitoring process execution and business performance. A key benefit of process implementation on a business process management server is the generation of events, or tracking signals, at each step of the process. These are filtered by the business process management system and aggregated into business measures and KPIs. Not only are the resulting metrics instantly viewable in management dashboards, but they can be used to generate real-time alerts and other triggered actions when performance exceeds preset thresholds.

Dashboards provide both strategic high-level views and detailed operational views of the business performance. High-level views allow executives and process owners to easily monitor the overall health of a range of business processes through graphical scorecards, tables, and charts, with business alerts indicating a situation requiring immediate attention. Operational views allow managers and analysts to drill down to KPIs associated with a specific process and see detailed breakouts by product, geography, work step, individual performance, or any other dimension specified for the metric.

4.3 Business process management and SOA

From a non-technical perspective, SOA is a set of business, process, organizational, and governance methods that help to create an efficient, effective, and agile business environment. From a more technical perspective, it is a way to standardize and improve application development and execution through the use of easily accessible, standardized, and reusable services. These services are independent of hardware and operating environments.

Most companies today are pressured by their customers and shareholders to drive growth by improving productivity and limiting cost in every aspect of their operations. But how can companies do that if they have rigid, expensive, and proprietary IT systems? It is a difficult task indeed. One of the most valuable projects a company can do today is implement systems with *flexibility*, that is, flexibility for challenges such as meeting new market demands and seizing opportunities before they are lost. To increase flexibility, a company has to look at its business as a collection of interconnected functions, or discrete processes, such as checking customer credit or authenticating a user. Then they can decide which of those functions are core, or differentiating, and which can be streamlined, outsourced, or even eliminated. If the company can mix and match these functions at will, or dynamically, in response to changing business conditions, they gain a significant business advantage. But to achieve this degree of flexibility in the business operations, the company needs an equally flexible IT environment. One way to do this is through an SOA.

SOA is also an applications framework that makes it easy to *reuse* and combine the discrete business processes defined for the business. Think of it as a mosaic made up of individual functional components that can be arranged and rearranged. With an SOA, the company can build, deploy, and integrate applications and link heterogeneous systems and platforms together across the organization.

With business process management, you distribute the business processes across the organization. With an SOA, you can integrate the IT applications to support your business processes. So you have business process management benefits of increasing effectiveness and efficiency, along with SOA benefits of flexibility and reuse.

IBM has developed an SOA reference architecture that provides a flexible, extensible, open standards-based infrastructure. This is a key added value of the IBM solution. The architecture specifies the required capabilities and services, and the defined interfaces that enable integration of the solution components. This is graphically depicted in Figure 4-6 on page 74. For more information about SOA, see the “Online resources” on page 427.

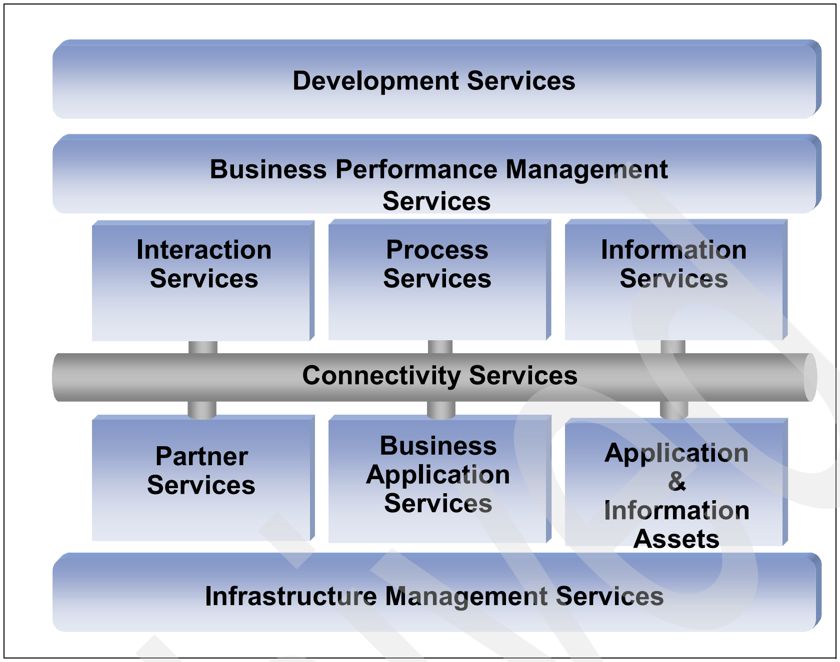


Figure 4-6 SOA illustrated

We have indicated that benefits of SOA are flexibility and component reuse. But, what are the forces driving this need for increased flexibility? Well, for one thing, consider economics. As the marketplace continues to globalize, new markets, new workforces, and new competitors are forcing companies become more flexible and able to adapt more quickly.

To support this, we see the cycle time between changes in business processes continually getting smaller. While you might have seen companies make significant changes yearly in the past, you now begin to see the same level of change on a quarterly, monthly, weekly, or even a daily basis.

While business leaders were focused more on cost-containment in the past, we are seeing that growth is now back at the top of the agenda today for the CEO. And, that growth demands flexibility so you can be more nimble than your competitors.

This is not to say that cost reduction has lost its importance. On the contrary, businesses are looking even harder for ways to make better use of their investments. There is more information available today than ever before, and companies need to be able to analyze it regardless of its location, format, or type.

And finally, SOA and the flexibility it brings are crucial for becoming what is referred to as an On Demand Business. An On Demand Business is one whose business processes, integrated end-to-end across the company and with key partners, suppliers, and customers, can respond with speed to any customer demand, market opportunity, or external threat.

SOA blends the best of all these concepts. But it is important to recognize that SOA is not the end of the road. It is the next step in the evolution of flexible infrastructures that can help you get much further down the road, and much more easily. This is graphically depicted in Figure 4-7.

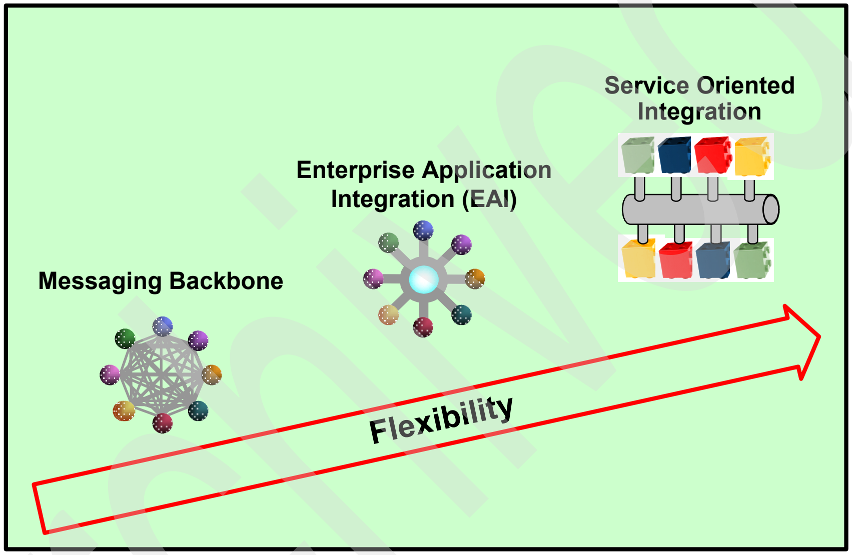


Figure 4-7 Transformation through time

There is actually an evolution depicted in Figure 4-7. We can recognize this by exploring some of the characteristics of each different connection:

- **Messaging Backbone:** This represents a point-to-point connection between applications. It is a very simple and basic type of connectivity.
- **Enterprise Application Integration (EIA):** Here the applications are connected via a centralized hub. It is easier then to manage a larger number of connections.
- **Service Oriented Architecture:** With SOA, the integration and choreography of services are through an Enterprise Service Bus. This provides flexible connections as well defined standards-based interfaces.

Evolving to this more flexible environment enables companies to bypass those integration and change-inhibitive barriers previously faced by IT and enable a more agile company that can easily morph to take advantage of leaps in technology and in the business environment.

4.4 Business process management tools and enablers

IBM has developed a business process management platform that enables the assembly of components encompassing business partner products and IBM foundation technologies. The platform includes a wide range of capabilities for modeling, integrating, connecting, monitoring, and managing business operations within an enterprise and across a value chain of trading partners and customers.

That business process management platform provides a set of associated interfaces for business partner plug-in components to customize the platform. These interfaces support facilities, such as:

- Business rules for dynamic process control and adaptive performance management
- Information management for analytics and reporting
- A common event infrastructure for the event-driven management of business and IT operations
- Workplace capabilities for business performance management visualization and collaboration
- Business services management for consolidated and dynamic resource management for aligning IT with business objectives

The IBM business process management platform enables IBM to efficiently assemble end-to-end solutions for specific business environments. The foundation capability anchoring the platform is the IBM extensible On Demand Business portfolio of strategy, framework, products, and technologies. This environment is depicted in Figure 4-8 on page 77.

The closed-loop framework depicts the processes in terms of modeling, assembling, deploying, and managing. With the flexibility and agility, this can be an ongoing process environment that enables fast and easy change and improvement. This is a great way to keep that competitive advantage.

Now, let us briefly look at the products that support each of those elements in the framework.

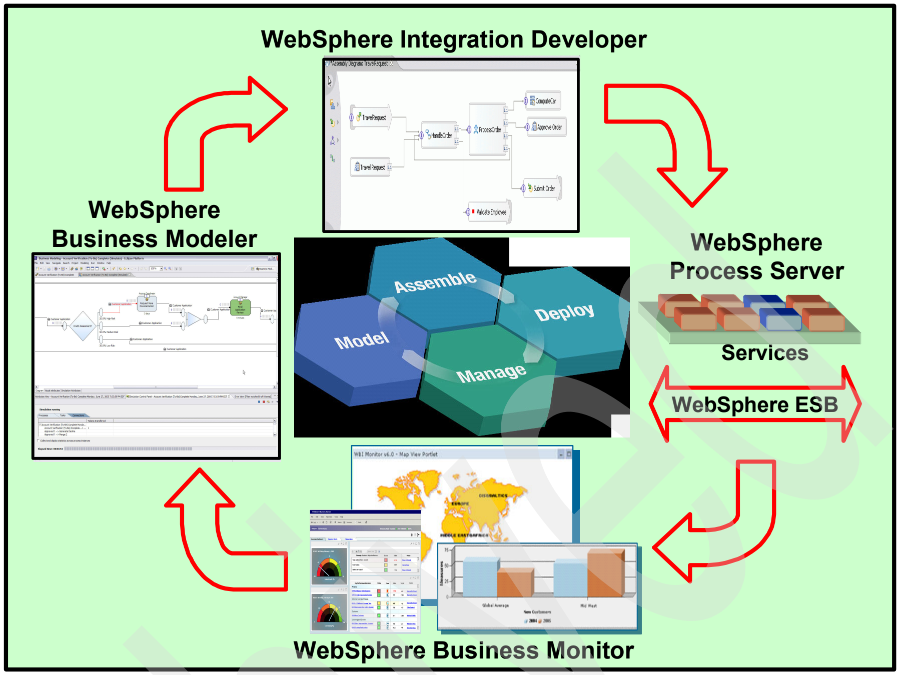


Figure 4-8 Business process management enablers

WebSphere Business Modeler

This is an Eclipse-based business process modeling tool that enables you to model, design, analyze, and generate reports for your business processes, integrate new and revised workflows, and define organizations, resources, and business items. It is designed for the business analyst to model, simulate, and optimize business processes before handing the model to IT for implementation refinements. Defining and modeling business processes is a key factor in improving business performance. For more detailed information about WebSphere Business Modeler, see 6.4, “WebSphere Business Modeler” on page 214.

WebSphere Integration Developer

This Eclipse-based toolset provides the answer to the integration challenges in the assembly of composite applications. It has been designed for IT developers and IT architects and links directly with WebSphere Business Modeler for seamless interaction between the different roles and organizations. To simplify and accelerate the development of integrated applications, this environment provides a layer of abstraction to separate the visually-presented components from the underlying implementation. For more detailed information about this product, see 6.6, "WebSphere Process Server and Integration Developer" on page 235.

WebSphere Process Server

This is a runtime environment for flexible deployment of business processes and orchestrates their execution. It makes plug-and-play of components a reality and provides the secure, robust, and scalable environment needed to deploy your mission-critical business processes. For business applications that require business integration using different technologies, it is the ideal platform. For more detailed information, see 6.6, "WebSphere Process Server and Integration Developer" on page 235.

WebSphere Business Monitor

As a Web-based client/server application, it monitors processes and process execution, measures business performance, and reports on business operations. This provides the real-time visibility into process performance, enabling process intervention and continuous improvement. It allows for visualization of key performance indicators, so that the health of the business can be monitored and arising problems can be pinpointed, allowing for immediate resolution. It also includes support for monitoring processes running in WebSphere Process Server. The information captured can help you identify problems, correct faults, and change to achieve a more efficient business process. For more detailed information on WebSphere Business Monitor, see 6.5, "WebSphere Business Monitor" on page 222.

4.5 Implementing business process management

In this section, we describe several considerations when starting a business process management project. The objective is to describe a set of tasks, rather than a methodology, that can help organizations as they begin their business process management implementation.

Determine organizational changes

This task enables you to understand the impact on your organizations. Changes in organizational processes impact organizational objectives and resources, such as employees, customers, and suppliers. Therefore, it is important that the task and changes be planned well and communicated well.

There are two types of organizational change:

- It is internally driven if the change is the result of an original idea. Such changes are likely to be inventions, innovations, or original process improvements. Improvements can affect product, process, or even worker compensation.
- It might be externally driven in response to actions, such as new legislation or by competition. It might also be the result of internally driven change of another company. For example, this often happen with vehicle improvements.

Determine process ownership

The process owner is the manager within the process who has the responsibility and authority for the overall process result. The process owner is responsible for the entire process but does not replace managers of departments containing one or more process components. Some activities of this role are:

- Determines and documents process requirements and secure customer concurrence.
- Defines the sub-process, including information used by the process.
- Designates line management ownership over this sub-process.
- Identifies, implements, and assures applications adhere to quality management principles.
- Ensures documentation of task-level procedures.
- Identifies critical success factors and key dependencies to meet the needs of the business during the tactical and strategic time frame.
- Establishes measurements and sets targets to monitor and continuously improve process effectiveness and efficiency.
- Reports process status and results.
- Identifies and implements changes to the process to meet the needs of the business.
- Ensures that information integrity exists throughout the process, including measurements at all levels.

Define the process input, output, and flow

The process requires one or more inputs and produces one or more outputs. It is best to begin by focusing on the *critical success factors*. The critical success factors are the inputs and outputs essential to meeting the mission of the process. These are the factors whose failure would cause the entire process to fail.

Identification of the inputs and outputs to be made from the point of view of the process suppliers and process customers:

- The *suppliers* to the process and the specific inputs required from each of them. A supplier is the entity whose work product (output of the process) is provided as input to the customer and must meet the customer requirements. The supplier can be either inside or outside the company.
- The *customers* of the process and the specific outputs required by them from the process. A customer is defined as the user to whom the output of the process is provided by the supplier, and whose requirements must be satisfied by the output.

The flow of the process must be accurately traced, and the movement through the process that converts inputs into outputs must be precisely displayed. This is done by working through the process, customer by customer. It is best to begin with critical success factors, which are the key customers of the process. For each identified customer of the process, the following procedure should take place:

1. Start with the output for that customer and identify the specific activity within the process that provides that output.
2. Identify the inputs for that activity, and trace them back to their sources.
3. Continue backtracking until at least one external input, and its supplier, for that output has been identified.

After completing the flow, it is necessary to ensure that supporting documentation is prepared. Supporting documentation is the base material for ongoing analysis and improvement.

Measure business process

Measurement of the process is used to achieve and maintain conformance to customer requirements. It requires continual monitoring of the status of the process to determine whether changes or improvements are required.

Measurements and objectives of business process management

The objective of business process management is a business process that is effective and efficient. An effective process produces output that conforms to customer requirements. The lack of process effectiveness is measured by the degree to which the process output does not conform to customer requirements. This is a primary aim of business process management if the processes are to be effective.

An efficient process produces the required output at the lowest possible (minimum) cost. That is, the process avoids waste or loss of resource in producing the required output and minimizes the cost of producing that output.

Another primary aim of business process management is to increase process efficiency without losing effectiveness.

What to measure

In every process, there are key activities in which favorable results are absolutely necessary for the manager of a process element to achieve the company goals.

These key activities are referred to as critical success factors. Each process normally has between three and five critical success factors. Once the measurements have been established for these critical success factors, there is time to further examine the process to devise measurements for the less important process elements.

Perform process analysis

Process analysis is a key step in the development of effective and efficient business processes. At this point, the mission of the process has been defined, boundaries and the scope have been defined and as well, an external influence on the process has been established. In addition, inputs, activities, and outputs have been defined. And lastly, a model of the various activities and tasks has been created. The “AS-IS“ (baseline) process has now been characterized and a level of understanding of the process has been established. The process is now ready for analysis, leading to further development and improvement.

Perform detailed analysis and revision

The information gathered from the interview creates a detailed picture of the process, and the team analyzes the process to see how it can be improved. During analysis, the team questions all parts of the process and considers alternatives. The objective is to identify the valuable tasks and make the revisions required to obtain the optimum process. This involves reviewing each task, revising as required, revising task sequence, eliminating unnecessary tasks, and eliminating causes of process failure.

Recommend and implement the improved process

When the analysis is complete, the team creates a new process flow to document the revised and improved process they are recommending. The recommendations should be accompanied with an explanation of what happened at each step, highlighting all changes from the original process. The team also summarizes the business advantages that justify the revisions.

Continuous improvement

The implementation of business process management requires continual improvements to the process. The first phases of implementation, as previously described, assure that:

- The process is defined and documented.
- Supplier and customer relationships and requirements have been identified.
- Quality measurements and measurement points have been established.
- Process simplification has been applied.

Once these basics have been satisfied, criteria must be established to assess how well the process meets the objectives. The analysis process leading to the improved process is on an iterative basis and should always be supported with action plans showing what the process owner must improve. Improvement is assessed relative to the criteria established for the process performance.

4.5.1 IBM tools for implementation

We have discussed many of the activities and process steps that can be included in a business process management implementation. In Figure 4-9 on page 83, we summarize those steps and the IBM products that can be used in their implementation.

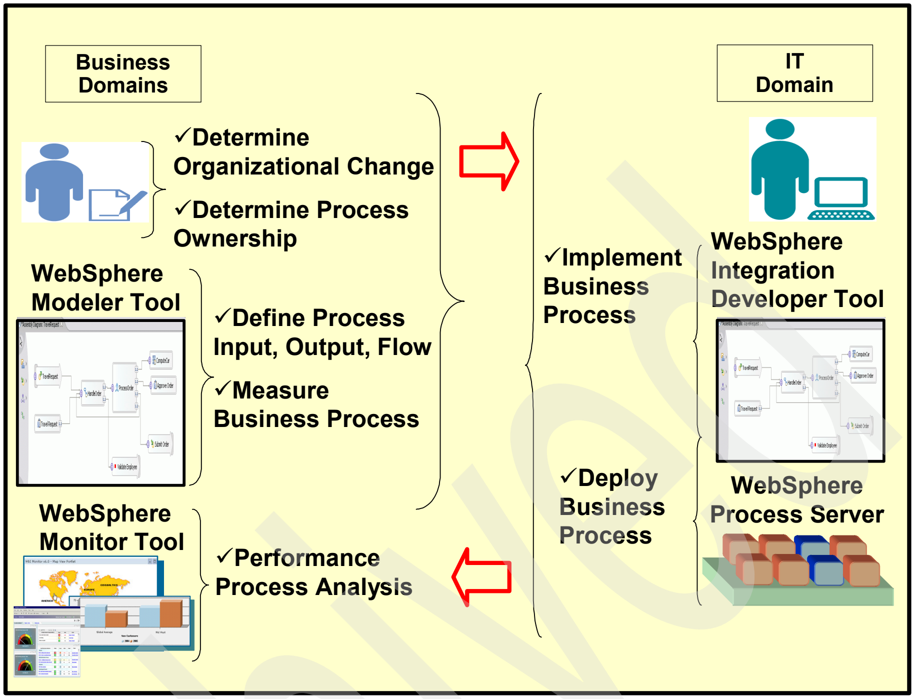


Figure 4-9 IBM tools for business process management activities

When the organization begins work on its business process management implementation, the first step is to determine what changes are required by the organization. Here the business managers work together to predict the organizational transformations that are required. For example, they define which processes have to change and who are the best persons to manage that change. After that, the business analyst group begins to define the process. For that, they use WebSphere Business Modeler. The result is a process model and all the elements in the process, including, for example, inputs, outputs, and resources. At this point, the business analyst can begin to suggest improvements to the processes through the use of simulation. Simulation enables the analyst to assess the performance of the process, generate statistics, and pinpoint potential areas of improvement.

The analyst now looks at the strategic objectives of the organization. Once defined, metrics can be defined with which to measure those objectives. The metrics can be defined as you define the process in WebSphere Business Modeler. The metrics are important because they help you to understand exactly how the business is running. With WebSphere Business Modeler, not only is it possible to develop the process, but the business analyst group can share all the business processes which are defined with all those in the organization. This activity helps to obtain feedback from those who work with the process and generates organization assets because now all the people in the organization can understand the business process in the same way.

When the business analyst group finishes, they export the model to the IT environment. The IT staff can then begin to work with the model and add technology attributes. To do this work, IBM has the WebSphere Integration Developer. When complete, the model is deployed with WebSphere Process Server.

It is important to note that the model was created for business people, and then it was implemented and deployed by, and for, IT people. All those people involved now work with the same business model. So what does that mean? It means that the business processes which are running on the server are the same as the model that was created with business people.

When the model is developed and deployed, it is now important to measure the results of those processes to verify they are meeting objectives. To do that, IBM has the WebSphere Business Monitor.

WebSphere Business Monitor depends on the business measures models for the appropriate monitoring procedure. These models are created in the WebSphere Business Measures Editor view, where you can specify the measuring points and event filters, and define the measurements, their correlations, and sources of the business data. When the business measures model is complete, it gets exported to WebSphere Business Monitor. It then recognizes the model to be monitored and the measurements to be captured from the incoming events. You use the Business Measures Editor to open the process models created in WebSphere Business Modeler and to create business measures models. For each business measures model, you can define the metrics and KPIs, event emission points, event filters, event composition rules, and situations that trigger specific actions at runtime.

The monitoring action objectives are there to improve your processes. So when, based on a metric, a problem is discovered, it is then possible to make a decision and change the process to correct the situation. These corrections occur in WebSphere Business Modeler. At that point, the cycle begins again.

4.6 Conclusion

Business process management is all about the *effective* and *efficient* management of business processes. We initially defined business process management around two different views, management disciplines and the technology platform. As a management discipline, it is focused in the business process and how the organization can be understood through the process. This helps management because it provides the opportunity to implement and monitor the end-to-end business process.

The business process changes are in models, models that represent the business of the company. As a technology platform, IT, through integration and enhanced technology, helps streamline the business transformation. One such transformation is through the implementation of SOA. SOA is focused on obtaining a technology that is *flexible* and *reusable*. Flexible because you can improve your business processes and can implement these improvements faster. Reusable because it is possible to use the same process elements in different processes to obtain business standardization and consistency.

Together business process management and SOA help to facilitate the next phase of business process evolution and that is going from merely automating repeatable processes to flexible automation of dynamic processes. This evolution is occurring because enterprises must compete more effectively by adapting to market changes faster, improving efficiency continuously, and streamlining collaboration across traditional silo departments. Modern business process management solutions, such as IBM WebSphere Business Modeler and Business Monitor, have helped to dramatically simplify the modeling, monitoring, and redesign of extremely complex processes. These business process management solutions make the process model a living representation of how organizations operate to deliver value and show how organizational operations can change to help increase that value.

5

Business Intelligence

In this chapter, we discuss the current stages of development of data warehousing and business intelligence solutions, and the extent to which these solutions can be used to improve the operational decision making process. In addition, for this redbook, we have a special focus on the integration of these initiatives with business process management.

Data warehousing is the process of extracting data from disparate applications (internal and external), transforming it into a more generic and meaningful format, and storing it into a consolidated data repository for business analysis. It requires a combination of methodologies, techniques, and hardware and software components that together provide the infrastructure to support this process, and the subsequent development and implementation of analytic applications.

Business intelligence is the process by which users can obtain accurate and consistent business data from the enterprise data warehousing environment, analyze this data from different business contexts, identify trends, variations and anomalies, execute simulations, and obtain insights about business problems or business opportunities that enable them to perform faster and make more informed decisions.

In the business world, most of the decision makers need access to accurate and timely information in order to achieve their business goals. They also need to have access to historical data to understand the past behavior of their business. But the focus today is about getting access to current transactions and business events in order to react quickly to new demands, market pressures, competitor movements, and other business challenges.

One of the challenges for the business intelligence solutions today is to minimize the delay between the time a business event occurs and the availability of the information required to take effective action. There are many factors that influence this delay, including the technology and architecture of the system used to collect, analyze, and deliver the information to decision makers.

To better understand these challenges, we explore some of the aspects of the evolution of the data warehouse and business intelligence systems, and the IBM recommended approaches to enhance information quality and minimize the gap between business events and information availability for decision makers.

5.1 The data warehousing evolution

Several of the first technologies and techniques used by many companies to improve the decision making process were the advent of the personal computer, spreadsheets, and personal databases. Skilled employees used these to obtain data from their operational systems and aggregate it into spreadsheets to produce reports and charts that could be used by decision makers.

This approach was inefficient, because it could not provide the information at the right time. There were multiple versions of the data extracted at different times. Multiple definitions of data elements and multiple transformation rules were used, making the data inconsistent, and even inaccurate, across many of these technologies. So, the company, and the decision makers, were faced with making decisions based on multiple different versions of the truth. The same data could be extracted by different departments at different times and using different transformation rules.

To try to resolve these issues, organizations began implementing centralized data repositories, or data warehouses. This approach held the promise of containing a historical, transformed, and consistent version of the data gathered from all business areas. There was some success, but it was not an easy task. And decision makers in specific business areas found it difficult to understand the data and data format and to access it in a timely manner. To address this issue, an additional layer called a *data mart* was introduced. The idea was for the business unit to organize the information relevant to their specific business area and manage it in their own version of a smaller data warehouse.

Because companies might not be able to justify the cost and time to implement an enterprise data warehouse, they decided to try a building blocks approach. That is, to build multiple data marts to satisfy the immediate demands of business units. The idea was to integrate them at a later time, basically building an enterprise data warehouse from the bottom up. This could result in less cash invested and yield fast results for certain business areas. However, this brought with it another set of issues. For example, since the data would eventually be loaded into the enterprise data warehouse, that meant it would exist, for some duration, in both environments. That was the introduction of data duplication, and all the issues that go with it, such as data inconsistency.

Technology improvements, such as reduced cost and increased speed and capacity of departmental servers spurred the growth of data marts. However, all this also gave incentives for business areas to move along faster. Many ignored the enterprise vision for faster implementation, resulting in the development of many isolated implementations, known as *data mart proliferation*.

In many cases, companies ended up at about the same place from where they were trying to get away. That is, they had redundant and inconsistent data. Or, as they became known, independent data marts. And, this was after spending a significant amount of money, time, and resources.

But now with the globalization of the economy, companies need to react fast due to changes in the market, such as the appearance of new and more competitors and the demand for new products. To remain competitive, companies needed access to more real-time data to be more flexible and dynamic in the fast moving marketplace. The historical and static data in the data warehouse was not sufficient by itself.

This environment called for a new layer of data in the data warehousing architecture. This new layer became know as the operational data store (ODS). The ODS, as the name implies, contains data from the just-completed business operational transactions. It was designed to quickly perform relatively simple queries on small amounts of data, rather than the complex queries on large amounts of data performed in the data warehouse.

5.1.1 Data mart proliferation

Some of the primary drivers for data marts are cost, speed of implementation, query performance, and control of the data. To most, this was justification enough for proliferating data marts. They decided that was still better than their current environment. Then as more data sources were available, those business areas needed access to those, too. And, they needed the transformation applications, known as extract, transform, and load (ETL), to incorporate the new data streams into their data mart, resulting in more cost, time, duplication of effort, and inconsistent data.

As you might imagine, over time this led to quite a tangled web for the enterprise, which we depict in Figure 5-1 on page 91.

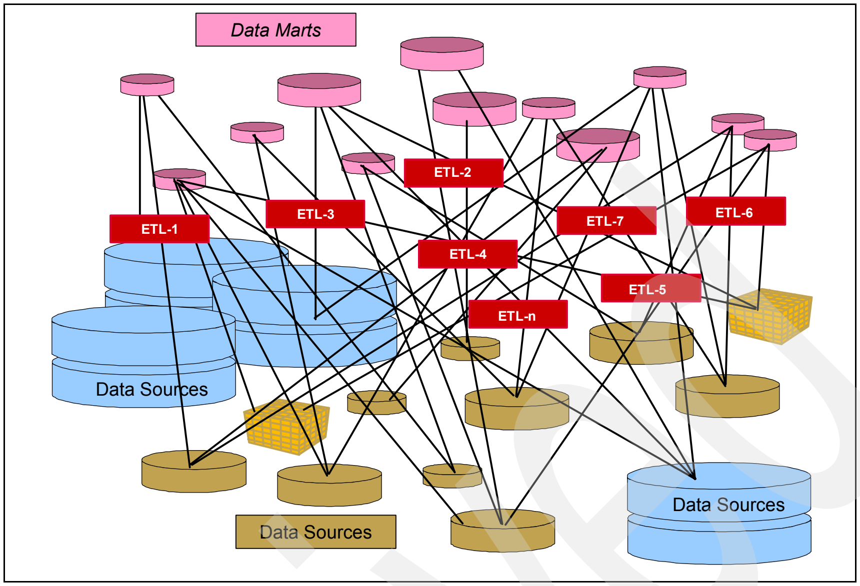


Figure 5-1 Data mart proliferation

And then there was more duplication from business decisions, such as mergers and acquisitions. In addition, many packaged application vendors insist on providing specialized data warehouse implementations for their own environments, an approach in opposition to the true spirit of data warehousing.

Although data duplication can provide some business benefits, it also creates a significant impact on the administration and maintenance of the environment. These data marts, warehouses, and operational data stores must be loaded and updated on a regular basis, leading to long-running and complex ETL processes that can easily affect the availability of the database to the users.

Data storage needs increased, although that impact might be offset by reduced cost of disk space. But it does not reduce the impact of the time required to populate each data mart. Even using a robust ETL process, there are still significant delays in information availability as more and more instances of the data need updating.

One of the biggest concerns with data mart proliferation is the cost of managing these multiple data stores and the increasing likelihood of inconsistency between them. Such inconsistencies defeat the entire purpose of data warehousing.

From a BI and business process management integration perspective, data mart proliferation adds more complexity and greater challenges in the automation of the operational decision process. Besides the administration complexity of the BI environment, the same data can be stored and maintained by different applications, which can easily lead to inaccurate information.

5.1.2 Independents data marts

Independents data marts lead to data inconsistency and inaccuracy, which hinders the execution of any business strategy. Data marts are built to satisfy the needs of specific business areas, with little or no thought given to integration with other business areas. They are silos of data with little or no enterprise control or perspective. These types of data marts are depicted in Figure 5-2.

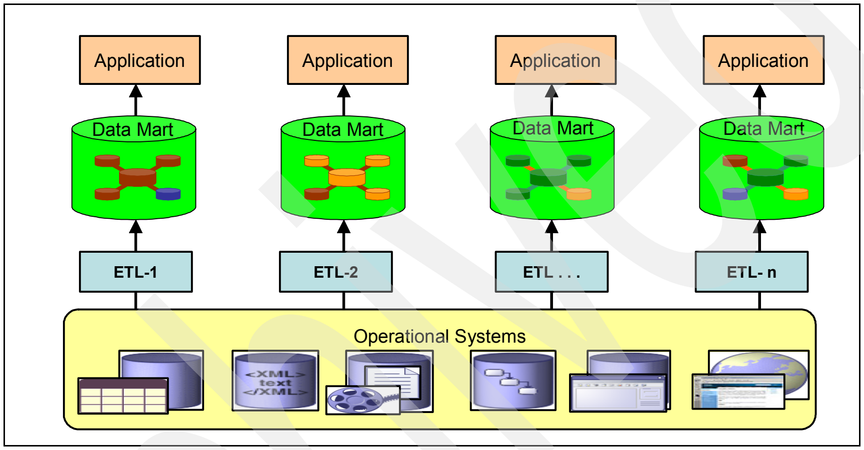


Figure 5-2 Independent data marts

Data for the data marts can come directly from one or more of the databases in the operational systems. And this is with few or no changes to the data in format or structure. This limits the types and scope of analysis that can be performed. This is the inconsistency and inaccuracy that we were originally trying to eliminate. It is still present in this type of environment.

Data can be extracted from the operational systems and transformed to provide a cleansed and enhanced set of data to be loaded into the data mart by passing through an ETL process. Although the data is enhanced, it is not consistent with, or in sync with, data from the enterprise data warehouse or other data marts. This is another issue impacting the credibility of reporting based on data marts.

Because these data marts are populated independently, they are all, by definition, populated at different times and using different transformation rules. Therefore, there can be no guarantee of data integrity.

In addition, the data for each data mart is physically stored on different, heterogeneous databases and servers, making any attempt at integration quite difficult.

This, in turn, can create challenges when you begin to integrate your BI and business process management environments. Untrustworthy information means untrustworthy decisions in the business processes.

5.1.3 Dependent data marts

Dependent data marts contain data that has been directly extracted from the enterprise data warehouse (EDW) environment rather than directly from the operational systems. Therefore, the data is integrated and is consistent with the data in the data warehouse. The EDW contains the atomic level of data granularity, and the data marts contain data that is of a higher level of granularity.

The data marts are built to support the business needs of a specific department or line of business. They help improve application performance because the data is highly aggregated and it is stored in data structures using star schema or snow flake models, which are optimized for query performance.

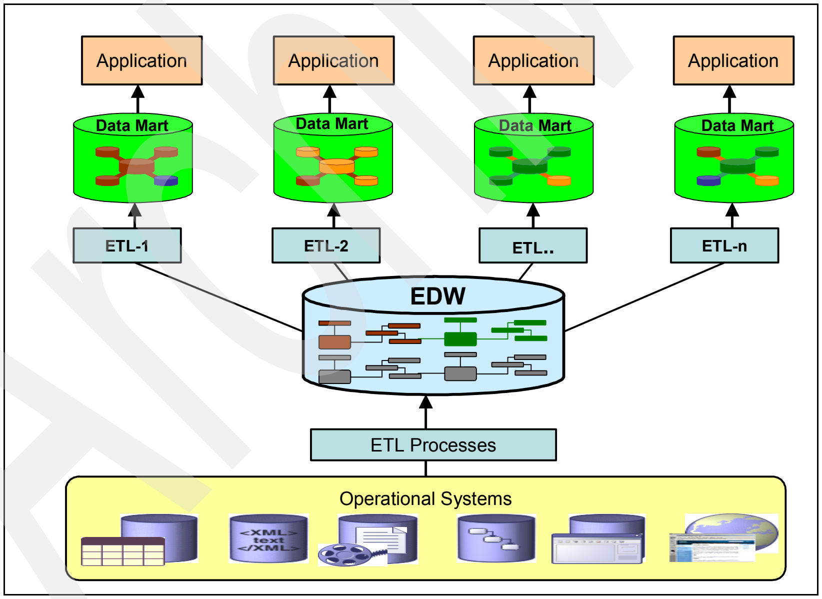


Figure 5-3 Dependent data marts

The EDW contains the single version of truth of the enterprise information. The dependent data marts that are populated from the EDW can reside on the same server and database with the EDW or on a separate database and server. In either case, there is still a need for ETL processes to populate the data marts. These ETL processes format the data appropriate for the specific needs of that data mart, aggregate the data, and also create new business metrics (calculated measures) when required.

Even though these are dependent data marts, they can still introduce a level of inconsistency, for example, if they are maintained on different time cycles. This also introduces issues, such as the latency of the information, because the maintenance might be required to be performed while the data mart is offline. And, there are still cost-of-storage issues because the same data might be maintained in a number of data marts.

5.1.4 Data mart consolidation

Along with the advent of data marts came their wide acceptance in the marketplace. They were seen as a fast and less demanding approach for implementing some of the support being demanded by departments and business units. In fact, many business units bypassed IT and implemented their own data marts. This led to a huge number of implementations, and what has been called *data mart proliferation*. And there were definite benefits realized by many companies.

However, along with those benefits, there were also down-side issues. For example, there were additional costs for administration and operation by both IT and the department or business unit, as well as significant increases in the expenditure for IT infrastructure (hardware and software). And with data mart proliferation, there can be processing delays that inhibit the timely delivery of data to the decision makers. Along with this came those intangible issues of inconsistent and redundant data, present in independently managed and controlled initiatives. This became a dilemma for IT.

Economic pressures are constantly forcing reductions in IT costs for products and services. This reduction does not only apply to direct costs of products and services, but also to costs associated with I/T infrastructure, administration, and operation of the I/T environment.

Consolidating the enterprise data is a major step in getting better control of the data. Having data managed from an enterprise perspective is the key to meeting the enterprise goals. It provides a single view of the enterprise, which can enable more informed decision making.

There are many other benefits in data mart consolidation, some are tangible and some are intangible. They center around reducing hardware and software license costs but also including the resources required to maintain the data marts. The intangible costs include costs such as the impact of data quality, consistency, and availability on decision making.

Of course, there are challenges in data mart consolidation. For example, it requires a good deal of effort to determine the preferred data consolidation technique and the effort to implement it.

Advances in hardware, software, and networking capabilities now make an enterprise level data warehouse a reachable and preferred solution. Here is a list of a few benefits of enabling data mart consolidation:

- Reduce IT costs by eliminating redundant IT hardware, software, data, and systems and the associated development, administration, and maintenance expenses.
- Improve decision making by establishing a high-quality, managed, integrated, and consistent source of analytic data.
- Establish a data management environment to support regulatory reporting for compliance with regulations, such as Basel II, Sarbanes Oxley, International Accounting Standards (IAS), and International Financial Reporting Standards (IFRS).
- Streamlined and efficient data delivery process that enhances the agility and responsiveness of an enterprise to new requirements and opportunities.
- Simplified environment that is easier and less expensive to keep and maintain.

5.2 Extending the data warehouse

In a traditional data warehouse implementation, data is physically moved from the source systems (operational systems) to the data warehouse.

The delays between the time that the data is produced in the source system and the time that it is available for access in the data warehouse by users is called *data latency*. This latency can vary, for example, from none to hours, days, or weeks. *Zero latency* means that as soon as the data is created by the source transaction system, it is also available for query in the data warehouse.

The business environment today requires an organization to react quickly to market changes. Access to data had become a critical component in the decision making process. In some cases, decision makers require real-time, or near real-time, access to the data.

Federation is a technique that allows the integration of data that is stored on heterogeneous databases and servers without the need to replicate (transfer and store) that data to the target system. That means users can have access to that data and data stored in the data warehouse at the same time.

The current information integration technology enables you to make this real-time data appear as if it is part of the data warehouse. Thus, you can avoid actually having to move the data. These techniques do not imply that it is a virtual data warehouse but simply extend the data warehouse capability by enabling it to access real-time information when needed.

IBM WebSphere Information Integration products provide the capability to federate data from a broad variety of data sources.

5.2.1 Right-time enterprise data warehouse

To compete effectively, companies are shifting their focus to reducing costs and increasing sales through business agility, responsiveness, and the timeliness of information. Businesses need to orchestrate activities among analytical, operational, and transactional systems. However, historically this has been very challenging and hindered by both technological and organizational obstacles.

Technology investments in reducing the latency within businesses are increasing, in particular those with emphasis on delivering information in a form that enables the business to react quickly, whether via automated or manual response. The timing of the delivery is not always actually real-time, but rather "*right-time*", that is, tailored to the appropriate response requirements of the information.

Traditionally, response to business events has been predicated on the ability to recognize these events within the semantics and scope of transactional and operational systems. The fundamental problem with this approach is that the source systems are limited in their ability to recognize events that might be interesting to the business, particularly when those events transcend the boundaries of a single system.

Business process management has enhanced the ability to recognize business events by providing a broader cross-application scope and some degree of shared semantics. Although new categories of business events can be defined and recognized using this technology, there are still categories of information, and thus potential business events, which are not considered.

Latency can also be defined as the elapsed time between a business event and an appropriate action or response to that event. So the keys to reducing latency are tied to improving the ability to recognize business events and the ability to respond to those events.

There are actually two related components working together:

- The first is a process issue manifested in the time lapse from when a business event occurs to when information about the event is recognized.
- The second is a process issue manifested in the time lapse from recognition to when an action is taken.

Analytical data has traditionally been created in a batch mode because the structure of the data is entirely different from transactional and operational systems. That is, it is optimized for specific types of analysis.

This structure can be complicated, its creation complex, and a very processor-intensive transformation task. In addition, the data is usually derived from multiple source systems that require additional matching and merging to arrive at a unified view. The data is also often viewed in multiple dimensions, for example, across time, to provide additional context. This adds to the processing requirements. All this processing has dictated that the analytical data be stored separately, and that the extraction and transformation processes are completed during batch cycles.

Beyond this data latency issue is the organizational separation of the groups that control the transactional and operational systems from the groups that control the analytical systems. The access knowledge and technologies for accessing analytical data does not exist on the process side of the organization, and the analytical technologies do not support the same standards.

Process-oriented technologies, therefore, do not have the ability to include analytical data transformation logic as a component of an event process or transaction.

In order to overcome these issues, the analytical data and data creation routines need to be published to the process-centric groups in technologies and standards which fit into their enterprise architectures and which do not compromise their real-time performance requirements.

5.2.2 On Demand Business intelligence

The value of analytical data in decision making has led to increased pressure to reduce the latency of this data. But, it is also very important to maintain accuracy and be consistent with the reports originating from transactional and operational systems.

Current ETL technologies allow the implementation of On Demand Business intelligence by reducing or eliminating the latency of data moving from the operational environment into the data warehouse and publishing that data in services that are consumed by applications and processes.

As data is created in source systems, an ETL engine can immediately transform it and populate operational data stores, data warehouses, and data marts on an event-driven basis, triggered from applications, using enterprise application integration, or via business processes.

This allows companies to take advantage of the best-in-class transformation and processing capabilities of the ETL tools, rather than endure the latency of batch processing. It also allows the data to be published as a service and to be easily consumed by applications and processes, and, without requiring application developers to understand the complex schemas and data sources that are associated with the data warehousing environment. We depict this in Figure 5-4.

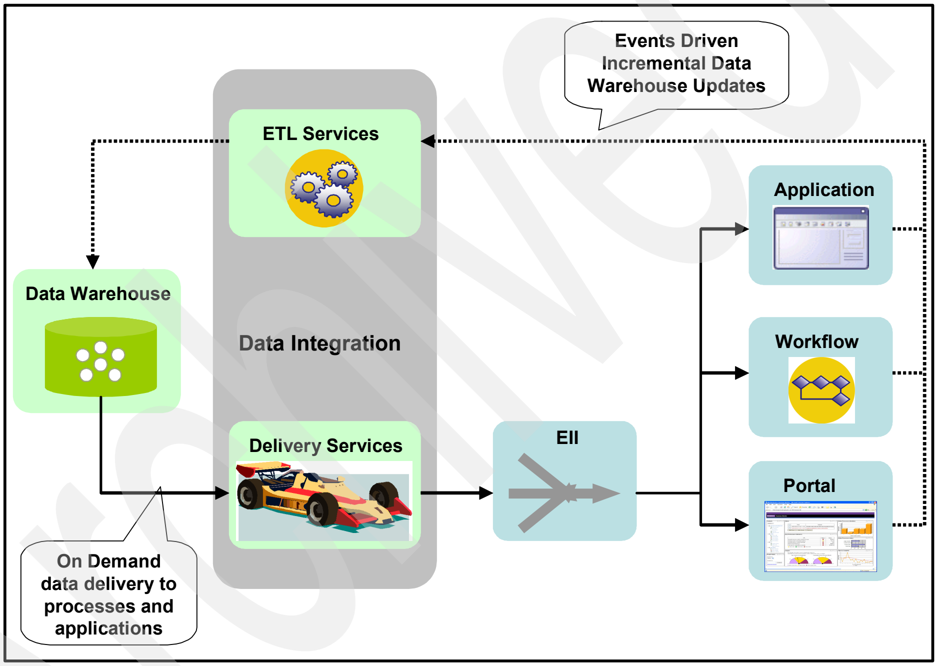


Figure 5-4 On Demand Business data warehousing

With less data latency across the spectrum, an ETL process can provide data integration services across analytical, transactional, or operational data stores. This allows analytical and operational data to be brought closer together, allowing automated operational decision making to also leverage the richer analytical data.

These data integration services can be easily called from any application, portal, or development project. They can also be called from enterprise information integration (EII) platforms to provide advanced data matching and transformation services to federated queries. Once in a familiar service-oriented structure, external applications and development teams are more likely to take advantage of these services to get the best available data and access it using a standardized approach.

5.2.3 Master Data Management

One of the first situations that companies discover when trying to reduce process latency is that their most vital data is often stored across many systems with little or no consistency. This forces development projects to go to extraordinary lengths to reach the correct sources of data and rationalize them into a single semantic representation. In most cases, it also means that applications and users rarely have a complete view of the enterprise data. This is commonly seen in customer marketing and customer service initiatives, where obtaining a single view of the customer still remains an elusive goal for many organizations.

Master Data Management (MDM) goes a step beyond On Demand Business data warehousing by creating authoritative sources of common reference data that can be used throughout organizational operations. The types of data targeted for this include data elements such as customer, product, and inventory.

Enterprises often choose this common data because it is accessed frequently across many applications, and the consistency of the data is very important to the business. Creating these master data stores improves the consistency and reliability of information for everyone and allows new development efforts to reuse proven standard access mechanisms rather than recreate them.

In a typical data warehouse implementation, the ETL process is used to extract source data from operational sources and apply transformations to make it more meaningful and more easily understood by users. The data is then loaded into the target data warehouse or data marts. Data standardization is also associated to the ETL process in order to guarantee that the same information that exists on different source databases is normalized before loading it into the data warehouse. The ultimate goal of the ETL process is to deliver accurate information to the users and guarantee that they have access to the same sources of data.

Business process management is gaining attention in many corporations to help enable them to optimize their business processes. Business process management solutions are implemented to automate, monitor, and control operational business processes. Operational decisions related to a process can be fully automated or they can be combined with human activities. The also can interact with business intelligence applications. For example, business process management solutions could obtain a customer score that is stored on the data warehouse and use it to automate a decision point of a specific process. A business process also can send notifications (alerts) that are related to a state or condition of an operational process that requires some type of action. Such notification could simply require further investigation and analysis by the decision maker before actions are performed.

Because data warehouses and data marts are built using business rules that are imbedded in ETL tools or scripts, it might be difficult to use those business rules to apply to a business process.

One of the key requirements of the integration of business process management and business intelligence is data integrity, especially with respect to the metadata. We depict this in Figure 5-5.

In this scenario, there is no synchronization between metadata in the data warehouse and the metadata in the business process management monitor database. An alert message was sent to the user regarding the high number of returns for a specific product (PROD01B), and the user needs to perform further analysis, using the data warehouse, to determine the root cause of this issue. Because the data warehouse contains normalized product data, it could be very difficult to relate the product analysis from the data warehouse with that derived from the dashboard, which was derived using data from the process monitor database.

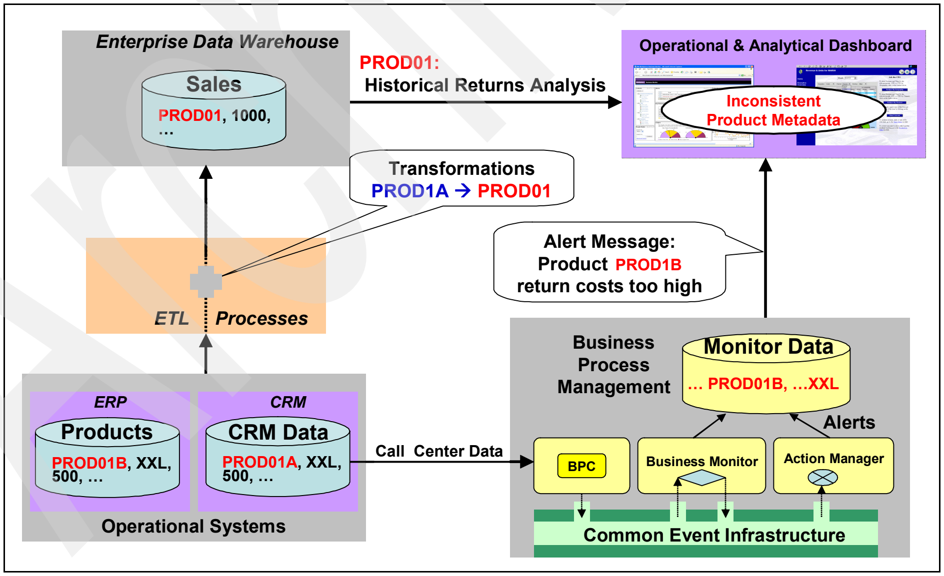


Figure 5-5 Business intelligence and business process management integration

Most companies today have multiple systems that require the same information. Often, each system utilizes its own data versions. And all too often, this leads to misaligned and inconsistent master data. This can lead to costly data redundancy and misleading analytics. MDM is an approach used to control, or eliminate, the proliferation and inconsistency of data.

With an MDM solution, the inconsistent data and misleading analysis between the data sourced from the data warehouse and the operational dashboard sourced from the monitor database could have potentially been avoided.

IBM defines MDM as the set of disciplines, technologies, and solutions used to create and maintain consistent, complete, contextual, and accurate business data for all stakeholders. It focuses on the concept of data objects, which represent the key business entities. Core master data objects include such elements as products, organizations, locations, trading partners, employees, customers, consumers, citizens, assets, accounts, and policies.

Because MDM utilizes SOA, it can easily be integrated with the process. And it also can be integrated with the ETL processes for the population of the data warehouse. As represented in Figure 5-6 on page 102, MDM is a primary source of product information that enables the synchronization between operational analyses and data warehouse analyses. A similar approach can be used to synchronize customer analyses that are sourced from both systems.

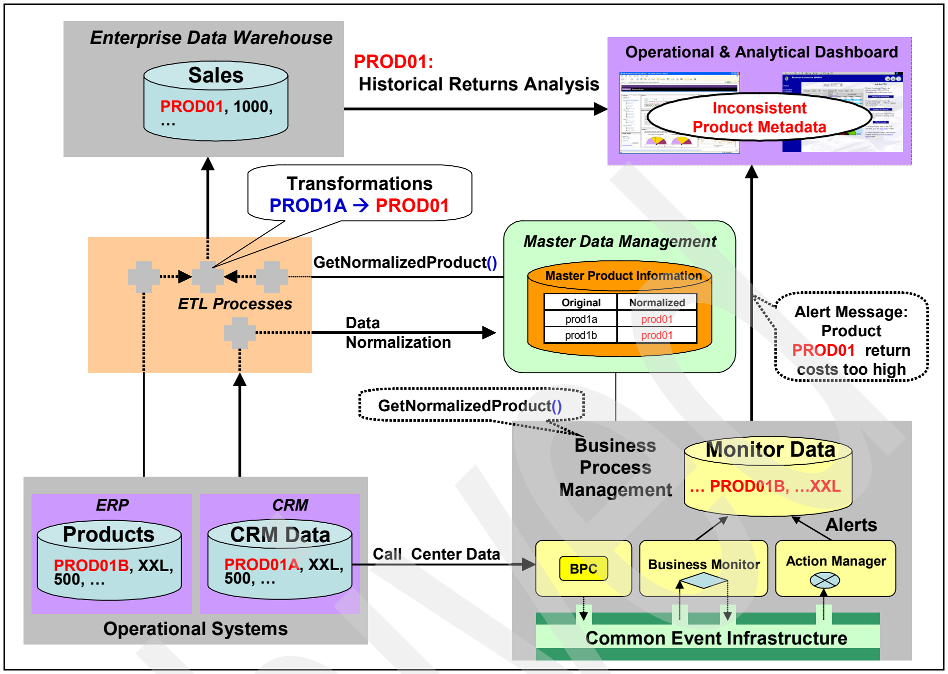


Figure 5-6 Integrated analytics with Master Data Management

If you are planning to implement business process management and perform integrated analysis with your existing data warehouse analytical environment, you should consider implementing MDM.

5.3 Layered data architecture for data warehousing

A *data architecture* is the overall design and structure of the data environment. It includes the software and hardware required to organize and store the business data in a format that can be easily manipulated and accessed by users. This design is supported by an enterprise data model that describes the nature of business data. The data model also contains data links and the relationships of the different business processes, and it provides consistent information.

A layered data architecture (LDA) is a design that organizes the business data in layers. These layers are defined based on the need for, and usage of, business data.

In general, there are three categories of decision making that require support from the data architecture. They are:

- *Operational:* These types of decisions require access to current business transaction data and determine the immediate actions to be taken during process execution. These types of decisions can be critical in enabling management to meet their business performance measurements.
- *Tactical:* Here the decisions provide direction at the departmental or functional area level. It is here that the plans for the enterprise strategy are developed and set in motion. The results of these decisions impact the processes executed and work performed at the operational level. Having access to more current data, but integrated with a historical context, enables management to plan, and replan, to better guide the operational departments toward meeting business measurements. This can require a level of workload planning and coordination among the operational business units.
- *Strategic:* These decisions, once thought to be long term, are also becoming more short term in nature. That is, in a fast-moving and ever changing business environment, long range strategy is becoming shorter and shorter in scope. However, these are the decisions that have an impact at the enterprise level and also need to be integrated with historical data. This is because here the coordination between the business units must be synchronized and directed toward the common enterprise measurements and goals.

A data delivery mechanism is used to transfer the data from source systems and make it available in the data warehouse in an appropriate and usable form that can be easily understood and accessed by business analysts. This process includes functions of data extraction, data cleansing, data transformation, data validation, data load, and data refresh. Data flow occurs between layers as the information is aged.

Traditionally, IT has viewed the layers of data only as separate layers in the data warehouse. It would then be mandatory to copy the data from one layer to another. This is depicted in Figure 5-7 on page 104. The data is physically instantiated in each layer. The users have direct access to any of the layers but only to one layer at the time.

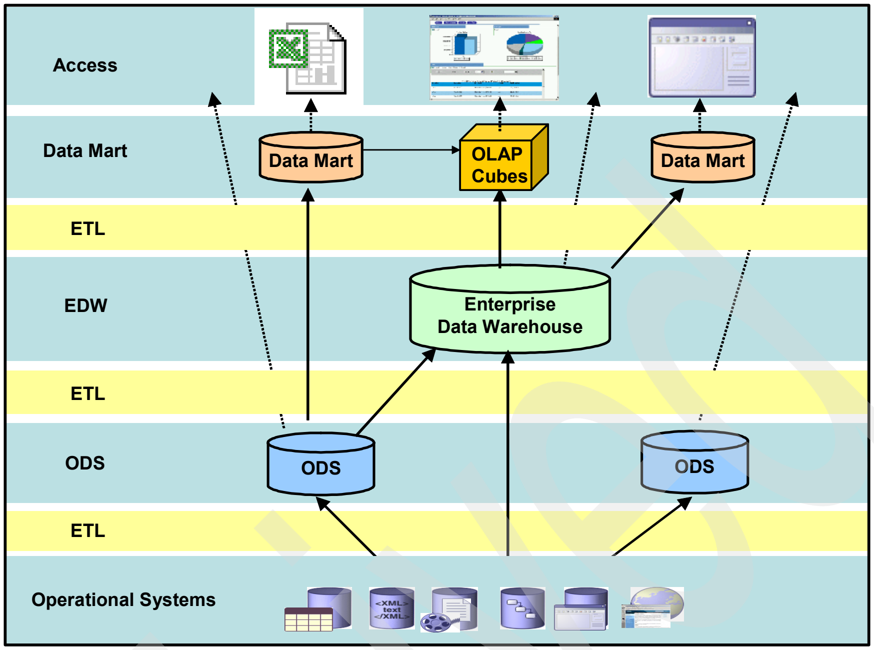


Figure 5-7 IT data warehouse layers view

From a conceptual point of view, business analysts see the data as a single layer in the data warehouse architecture. Regardless of the nature of the decision process, or the source of the data or the storage mechanism used, they basically need to have transparent access to trusted data at the time it is needed for decision making. The current stage of technology evolution allows the implementation of an architectural framework that delivers consistent, timeless, and quality information. We define this architectural framework as the layered data architecture. It is represented by three layers as depicted in Figure 5-8 on page 106.

The data assets layer is comprised of a sophisticated data delivery mechanism that is responsible for maintaining the data flow available to users. This data is organized on a application neutral data model that contains instantiated data structures as well as links to external data when physical instantiation is not required.

The data delivery mechanism includes batch processing for historical information and large volumes of data. Enterprise application integration (EAI) processes enable near real-time synchronization of changes from source to target systems. Federated access is enabled by an enterprise information integration (EII) bus for information that needs to be instantly available to users. This is particularly critical when the information is stored in multiple sources, and the real-time data is required but no data transfer from the source system to target system is possible.

The business access layer contains views to the application neutral data model in the data assets layer. These views are required to have the data formatted in a way that makes it easily understood by the users. The key benefits for this layer are the flexibility, quality, and availability of the information to decision makers.

New applications can be prototyped without the need of complex processes to copy and move data to a new repository. The same applies when additional functionality for an existing application is required. The information is available for users when they need it, because this layer does not require that the data be instantiated into physical structures and so there is no additional delay in making it available. There is also no need to create duplicate copies of the data every time a new application is required. But the most important benefit is the quality of the data. All users have access to the same base data, and so the information is consistent across departments and the organization.

The performance layer contains additional physical data structures as required to improve performance for applications. These data structures can be implemented, for example, by using materialized query tables (MQTs) or by simply creating additional database tables. Such a layer is required because the business views might need to manipulate and aggregate large volumes of data, which could impose delays on the query processes. The MQTs precompute the data and can, therefore, dramatically improve performance. The beauty is that the queries and applications do not need to be aware of such objects. This is because the database optimizer rewrites the incoming queries to access the MQTs whenever one is available that can provide the same results.

.

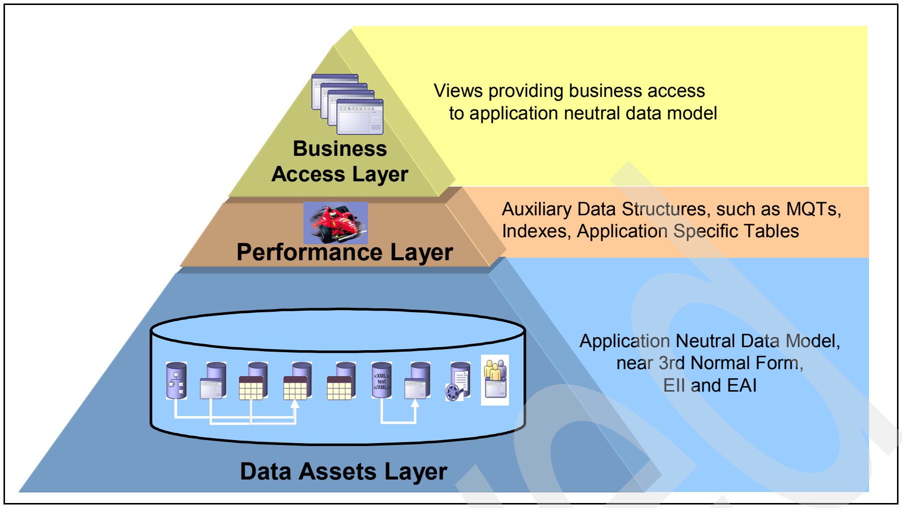


Figure 5-8 Conceptual Layered Data Architecture (LDA)

The LDA is the vehicle to improve information quality and availability while keeping the flexibility for line-of-business and departments to build their own applications. It also contributes significantly to the reduction of information technology costs because the data is only materialized in the upper layers when it is needed for performance reasons or some other particular business reason. But the key benefit of the LDA is that it enables direct access to the data in a format that is easily understood by the business users at the time that it is required. It also removes the requirement to copy data between different layers of the enterprise data warehouse environment.

Some enterprises today have already begun the process of consolidating their data environments, reducing the number of copies of data to a more manageable level. It is important to note that consolidation of a data warehouse environment does not necessarily mean simply collapsing it all to a single, highly normalized database in which each data item exists once and only once. Some level of redundancy is not only necessary, but can be desirable in a data warehouse.

For example, experience in the use of highly normalized data warehouses shows that some tables are joined in the majority of queries that use them. Similarly, summaries and aggregations that exist at the data mart level are constantly used and reused. Precomputing and storing such joins and aggregates both improves query performance and provides the best balance between processing and storage resources. Such consolidation depends, of course, on the availability of powerful processors and specific database features to provide adequate query performance.

Recently, many large data warehouse customers have begun testing the approach of combining multiple data marts into a single, centralized mart, then using views and MQTs to create virtual data marts within the data warehouse. When using views in this environment, MQTs provide a powerful way to bridge the performance gap seen when accessing the normalized EDW directly via views.

This movement toward increased data consolidation gives rise to a more fluid view of the overall information environment, as shown in Figure 5-9. This diagram emphasizes several key points.

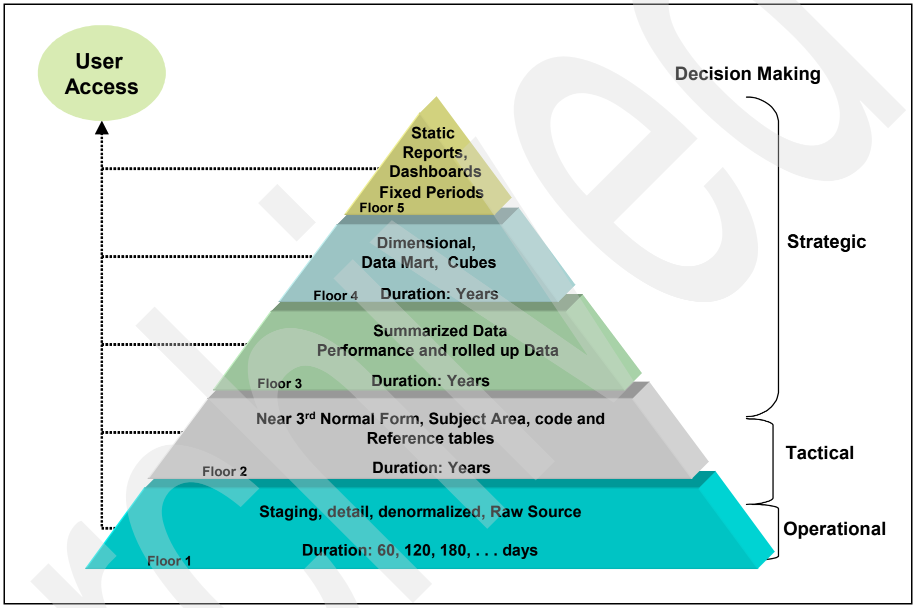


Figure 5-9 Floors of the information pyramid

For example, it conveys the essential unity of the information used by an enterprise, from detailed transactional data to consolidated and summarized data aggregates.

The business analysts correctly see the layers simply as different views of the same data, although individuals might focus more on a particular floor to do their specific jobs. To emphasize this difference, we named these divisions floors, rather than layers. While some data copying might continue between the floors, this approach is no longer the only possible approach. There are a number of approaches that enable integration of the data in the enterprise, and there are tools that enable those approaches.

The data on the different floors has different characteristics, such as volumes, structures, and access methods. Based on that, you can choose how best to physically instantiate it. However, given appropriate technology, you could also instantiate many of them together in a single physical environment.

Floors 1 to 5 of the information pyramid can be mapped to the layers in existing data warehouse architectures, because these layers are based on the same fundamental data characteristics. However, such mapping might mislead the essential unity of the views of data and perhaps should only be used for migration purposes. The preferred view is one of an integrated enterprise source of data for decision making, and a view that is current, or real-time.

In addition, the diagram in Figure 5-9 on page 107 emphasizes that users can have access to each floor. They could request information from a single floor or from multiple floors.

5.3.1 DB2 UDB and the layered data architecture

DB2 UDB is ideally suited to support a comprehensive and layered model of data usage that spans the full spectrum from transaction-consistent read/write activities to highly specialized analysis functions that require dedicated read-only data. Some of the DB2 UDB features that support this include:

- MQTs to provide specialized data structures to aid performance, specifically at floor 3 (supporting a virtual floor 4) and at floor 4 to support star schemas.
- Parallel SQL operations, such as MERGE (UPSERT), SELECT over INSERT/UPDATE/DELETE, declared global temporary tables (DGTTs), unlogged operations, and the cross loader enable building the upper floors with this approach. Such functions are critical when building new tables from existing ones for added application needs, for example, building parts of floor 4 or 5 from lower floors such as floor 3, which has the normalized subject areas for the entire business.
- Workload management functions, such as the DB2 Query Patroller and Governor, address new challenges of mixed workloads. This includes workloads containing online transaction processing (OLTP) type queries mixed with decision support system (DSS) type queries.
- Tablespaces allow *sand boxing* in the larger environment that was previously done in data marts. Sand boxing allows application development within the production system. In addition to workload management controls, disk quotas prevent an application development team from taking over the database.
- Online utilities support an environment in which different parts of the system have very different maintenance windows. In the multi-floor architecture, the whole system might have a very small maintenance window, or ultimately none at all. In this case, online utilities are key.

5.3.2 Data warehousing: The big picture

To remain competitive, organizations need to improve profitability, reduce costs, and respond faster to competitors' moves, market opportunities, regulatory changes, and day-to-day process exceptions. Such requirements can be directly influenced by using the data in the data warehouse.

The data warehouse functions go beyond a repository of historical business information used by for front-line business decision makers for strategic and tactical analysis and decisions. It also can incorporate current and transactional data that can be used to improve operational business decisions. The enterprise is becoming more and more unified and integrated. The layered data architecture supports this and enables it just like the ongoing integration of BI and the business processes. An integrated enterprise strategy is key, and the more robust hardware and software to support it are here today.

Analytic applications are also used more frequently to deliver data and initiate corrective processes and activities in right time. Activities that require immediate action for a specific operation of the business might include system-generated information or guided analysis sourced from a data warehouse. Information from specific and current transactions could be used to trigger alerts and support strategic decisions to avoid risk exposure and assure compliance.

Looking at the big picture, as represented by Figure 5-10 on page 110, the data warehouse can be integrated with an SOA and deliver information to consumers, such as processes and applications. Here, the different data integration technologies are applied to assure that the information is timeless and consistent across the enterprise.

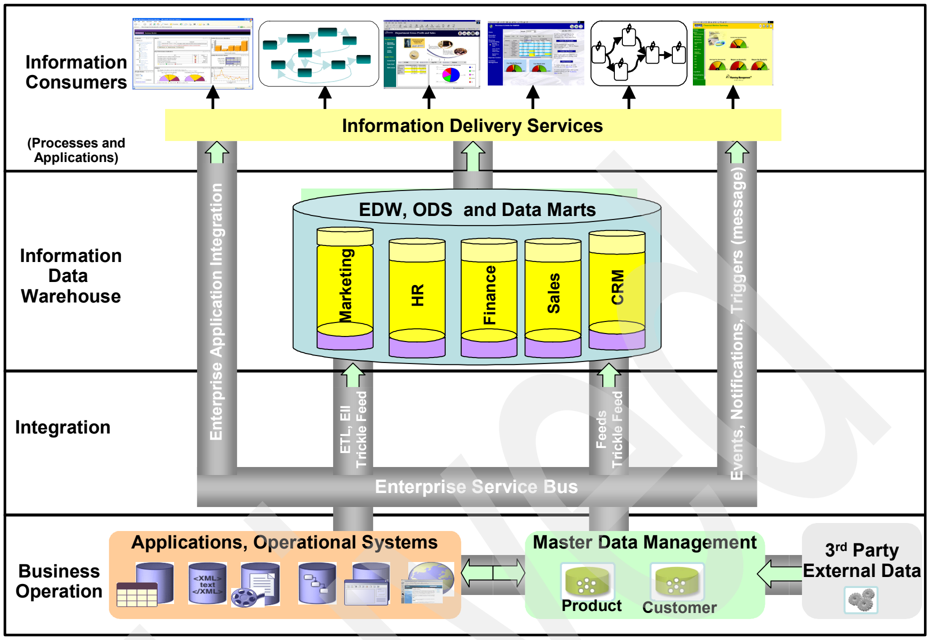


Figure 5-10 Right-time enterprise data warehouse

Certain high volume transaction systems require specialized applications to automate tasks that require operational decisions. For example, an insurance company could use an Internet application to explore a new and large market of potential customers. Traditionally, the process of generating an insurance quote requires an assessment of the potential risk factors for the customers. The current stage of development of the technology allows an insurance company to implement a model-based underwriting system and deploy it into a process server that can provide immediate responses to Web policy applicants. This process can utilize information from a data mining application, for example, a score, to perform a risk assessment, and, based on a predefined threshold, accept or reject the application. Such a process could lead to increased revenue, cost reduction, and lowered risks.

Similar systems could be applied to automate the operational decision process in other industries, not only for fully automated activities, but also to improve the decision making processes of activities that require human intervention. For example, a Call Center operator could potentially use information from a data warehouse for promotional offering personalization.

In order to succeed in the integration of operational and strategical systems, you need to consider the implementation of a enterprise data integration policy. You really want to avoid seeing different results from reports originating from transactional and operational systems compared to results seen in the analytical reports. Data governance should also be in place to protect and avoid exposure of critical information.

5.4 Data integration

Although the discussion in this section centers on technologies and architectural opinions in different areas of data integration, there are commonalities that cannot be bypassed, regardless of the method you use. All integration solutions, whether they are created with ETL, database federation, or application integration, must deal with disparate applications, different data formats, different standards, and the inevitability of change.

To be successful, you need to execute integration projects with a level of consistency and governance to address some of the key challenges. For example, lack of confidence in the correctness of information that is being delivered to the organization leads to poor and delayed decision making. Not having interface design principles and common formats for business object definitions leads to point-to-point application approaches and prevents reuse of already defined interfaces.

Technical challenges for all integration projects involve correct formats and semantic definitions to be able to merge two or more disparate repositories of application data. A methodology to document technical items, such as record definitions, structures, interfaces, and flows must be put in place to ensure a level of consistency and confidence in the information being delivered. An Integration Governance Model that is supported by all organizations involved in the integration process must control those definitions.

5.4.1 Extract, Transform, and Load (ETL)

ETL tools have long been the workhorses of data integration. They were created to extract the information, transform it into a consolidated view, and then load it into a data warehouse in a batch mode. The data volumes involved are generally large, the load cycles long, and information in the data warehouse can be a day to a week old. For synchronizing data across operational systems, operational data stores, which enable the real-time update of information, were created.

Designed to process very large amounts of data, ETL provides a suitable platform for improved productivity by reuse of objects and transformations, strict methodology, and better metadata support, including impact analysis.

The problem with the ETL approach is the need to physically move large volumes of data from source systems to multiple consolidated data stores, including the data warehouse, distributed data marts, operational data stores, and analytical multidimensional databases. While these consolidated data sources continue to be important to organizations, latencies and inconsistencies might still be present in such architectures.

5.4.2 Enterprise Application Integration (EAI)

Batch ETL solutions are, in general, incapable of meeting the real-time integration needs of the new breed of online systems, because the information can be a day or more old. While ETL tools continue to serve a valuable function in organizations, their functionality is more and more often replaced by integration.

The newer EAI solutions came along and solved many of the data latency problems by synchronizing changes across systems in real time. EAI focuses on the integration of data among a collection of applications or systems. As data is changed in one system, the change is propagated to other systems of interest, usually via asynchronous messaging. Application integration, though required by business functions, is primarily the domain of an IT organization. The responsibility of the EAI systems is to keep the systems within an organization synchronized with each other.

The issue with some application integration platform products is scalability and transformation power. EAI less adequately addresses the need to aggregate and consolidate data and information across the enterprise. EAI can effectively move data among systems in real time, but it does not define an aggregated view of the data objects or business entities.

5.4.3 Enterprise Information Integration (EII)

EII is the integration of data from multiple systems into a unified, consistent, and accurate representation appropriate for viewing and the manipulation of the data. Data is aggregated, restructured, and relabeled (if necessary) and presented to the user.

Information integration targets the requirement of dealing with data from multiple systems without moving it to an integrated source. This characteristic differentiates EII from ETL and EAI, which require all the data to be identified and moved prior to user access.

For example, a customer service representative needs to answer a customer question requiring data from multiple heterogeneous sources, in real time.

This requires the ability to create a query that can access distributed data sources as though they were a single database.

Providing a unified view of data from disparate systems comes with a unique set of requirements and constraints. First, the data should be accessible in a real-time fashion, meaning that it should be accessible directly from the source as opposed to a copy of the source. And, the semantics, or meaning, of data need to be resolved across those disparate systems. This is because those disparate systems might represent the equivalent data with different labels and formats. However, that would require some type of data correlation by the user. These types of duplications and inconsistencies should be removed, their validity checked, the labels matched, and values reformatted by the system, not the user.

The challenges with EII involve governing the use of a collection of systems in real time and creating a semantic layer that can map all data entities in a coherent view of the enterprise data.

Federated semantic layer

An element of any data integration project is to provide a consistent view of the data across the enterprise. That means enterprise standards for data entities should be created and maintained.

Every approach implements the same standards, even if in different ways. Federation and, in particular, WebSphere II rely on a common metadata layer to join elements and resolve discrepancies across multiple data sources. With WebSphere II, this semantic layer is implemented using database definitions, such as nicknames, views, stored procedures (SPs), and user-defined functions (UDFs). The main challenge when defining such a layer is to keep it orchestrated with the enterprise definitions. This coordination is a task reserved to DBAs, who must be able to manage the changes in the standards while maintaining the integrity of the user views of the data.

Remote access governance

One of the dangers of implementing an EII infrastructure is to attempt to create a virtual data repository where several data domains are distributed across distinct repositories. In many cases, ad hoc use of such a system can create impossible workloads that can impact performance and capacity of the sources involved.

It is imperative that a federated solution have access to current data in remote systems but you must be aware of the potential impact that federated queries have on such sources. Special attention must be given to operational systems, which are not designed to support analytical queries.

Federated governance can be achieved by design or by using workload management (WLM) tools. A definition of a semantic layer is imperative for a successful federated implementation. This layer can be used not only to solve discrepancies across data sources but also to provide a level of governance to the federated queries by limiting the domain access to remote sources. For example, a nickname can be created on a very large remote product table but access given only to a few categories, by using database views.

WLM tools, such as DB2 Query Patroller (QP) and the DB2 Governor, can also be used to provide an additional level of governance for federated queries. WebSphere II relies on cost-based optimization to create federated query plans. The cost is affected by network and remote server capacity. By issuing queries that might significantly impact those two types of resources, the query cost increases. QP strategies can be defined to control such workloads. The DB2 Governor provides an additional control level in case runaway queries are not captured by QP.

5.5 Scaling a DB2 data warehouse

In this section, we describe some of the key concepts and components that enable DB2 to provide consistent and fast performance for a data warehouse. We explore some of the technical aspects of the database engine as well the hardware configuration characteristics that are required to provide scalability for analytical applications.

Why is scalability important in BI and business process management integration? In a typical data warehouse environment, there is a hybrid mix of concurrent queries executing during any given period of time. There are queries of long duration which require access to large amounts of data (historical information), queries of short duration that only need to retrieve a small number of rows, and queries of medium duration that require various amounts of data. From a response time expectation, a user might be satisfied with waiting several minutes to get the result of a query that can process millions or billions of rows. However, a user gets impatient if there is a need to wait the same amount of time to get results back from a very small query. For this reason, it is really important that the database engine supporting the data warehouse environment has the capability to control workload and give priority for queries that need to be processed sooner.

From a business process management and BI integration perspective, the amount of data that flows from the data warehouse to feed a business process is really small, such as a single row with customer scores. Such a request is really expected to be processed very fast. This is because the query really requires a minimum amount of processing resources, and because the characteristics of the process are really transactional, requiring a real-time response. Another important consideration is the volume of new transactions that a business process management environment can add to a data warehouse environment. Here the database engine really needs to be able to scale to support a very large number of small transactions without impacting the overall performance of the existing BI applications.

In the next sections, we explore some of the important features and characteristics that enable DB2 to support processing a large number of concurrent requests while keeping the response time consistent.

5.5.1 DB2 shared nothing architecture

A DB2 database can consist of one or more database partitions. Each database partition is, in essence, a mini-database. It has responsibility for its own (and only its own) data, logs, data locking, and other essential elements that comprise a database. For this reason, the concept of a multiple-partition database is often referred to as a *shared nothing*, or massively parallel processing (MPP), database architecture.

As depicted in Figure 5-11 on page 116, a DB2 database can consist of several database partitions. Each database partition contains its own resources, such as data, logs, caches, and locking management. Parallel processing occurs on all partitions and is coordinated by the DBMS. The communication between partitions is handled by a DB2 component called Fast Communication Manager.

The *data partition* is a DB2 UDB database partition that is dedicated to storing partitioned database data from one or multiple tables within the same database.

The *coordinator partition* is a DB2 UDB database partition to manage user connections and coordinate queries. It is responsible for consolidating the result set of a query that spans multiple data partitions for parallel data retrieval.

The *catalog partition* is a DB2 UDB database partition that contains the DB2 system catalog tables. In general, the catalog partition and coordination are together.

It is important to consider in this architecture that each database partition contains dedicated storage (disk). Because each database partition has access to exclusive disk devices, a high level of parallelism can be achieved without impacting the performance for the I/O (reads and writes). For example, when a user sends a request (query) to select data from a table, DB2 distributes the query to each data partition to retrieve the required data. All database partitions work in parallel to retrieve the data. But because each partition contains its own disk, there is no competition for I/O (disk access). Each data partition sends the partial result back to the coordinator partition that is responsible for consolidating and sending the complete result to the requester (user).

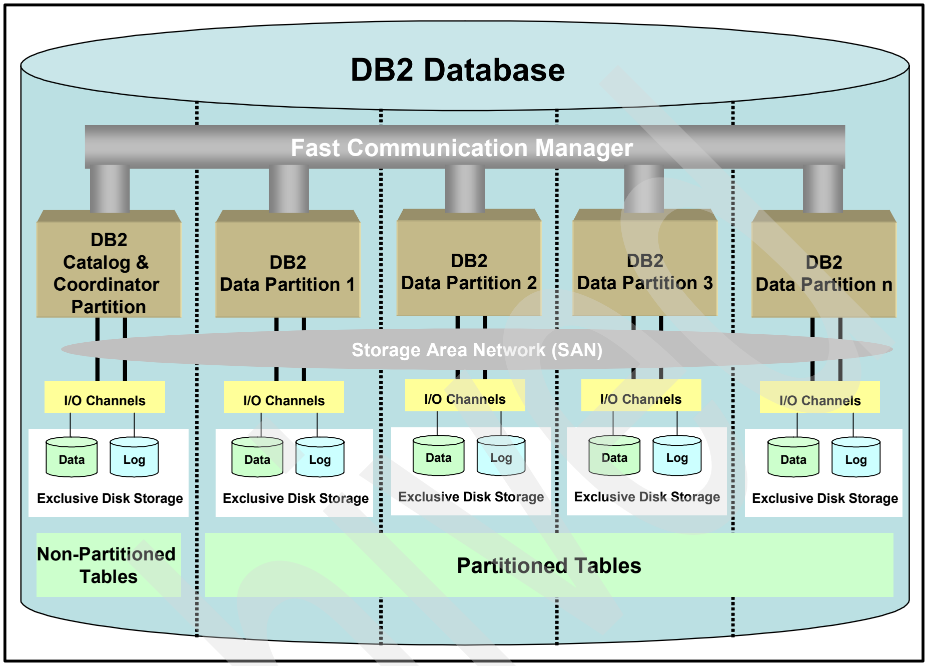


Figure 5-11 DB2 shared nothing architecture

A database (software) architecture that uses a shared nothing approach can be hosted on a single symmetric multiprocessor (SMP), where the partitions of the database would all reside on that server. The shared nothing database can be used in an SMP environment. By adding additional processors, memory, and disk to an SMP, the processing capacity can be increased.

5.5.2 DB2 Balanced Partition Unit (BPU)

The balanced partition unit (BPU) is a logical rather than physical concept and primarily refers to the resources that are required to service a DB2 UDB database partition. Because in most data warehouse architectures the database (DB2 UDB) is the only software component capable of supporting data partitioning, the BPU primarily refers to DB2 UDB.

Figure 5-12 on page 117 depicts a DB2 BPU, which is a combination of resources such as CPU, memory, a database partition, DB2 agents, memory, communications subsystems, I/O channels, and dedicated storage.

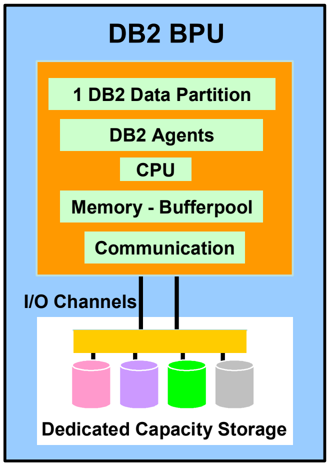


Figure 5-12 DB2 balanced partition unit

Even if it is possible to dedicate some resources to a BPU, most of them can be shared across different BPUs. However, we strongly recommend that the storage (physical disk devices) component be dedicated to each individual BPU.

Like the DB2 database partitions, the BPU also can be conceptualized by usage:

- **Data BPU**. A DB2 UDB database partition dedicated to storing partitioned database data.
- **Coordinator BPU**. The DB2 UDB coordinator partition manages user connections and coordinates query execution.
- **Catalog BPU**. This is the database partition in which the DB2 catalog resides. The catalog partition is not dedicated, it is in a coordinator partition (BPU).

Although a single DB2 UDB database partition can service all these types and uses, it is a good practice in database design to use separate database partitions for each use.

The BPU is most useful at performing data warehouse sizing and capacity planning. For example, assume that a BPU (DB2 UDB database partition) is capable of supporting 125 GBs of raw data. Extrapolating for a 4 TB of raw-data data warehouse, this results in a requirement of approximately 32 BPUs.

Also, because DB2 balances the data and resources equally among the database partitions, when the system requirements for a BPU have been sized (in terms of CPU, memory, and I/O), this information can be used to determine the overall hardware configuration for the database.

5.5.3 DB2 database topology

DB2 supports multiple topologies for a partitioned database implementation. As Figure 5-13 demonstrates, there are three possible implementations:

- *MPP:* This environment contains many servers, each of which has a single CPU and operating system. The servers that are part of the MPP environment are interconnected via a high speed network. The database is portioned across all servers, but there is a single database partition per server. Even if all servers can be connected to a type of network storage, each database partition contains dedicated disk devices (no shared disk).
- *SMP:* This environment contains many CPUs in a single server with one operating system. The database contains multiple database partitions within the server. Each partition shares memory and CPUs, but each partition contains dedicated disk storage.
- *Cluster:* Here there are many servers where each server is an SMP server. The database is partitioned within all the SMP servers that are part of the cluster. The database partitions share memory and CPUs within the SMP server, but each partition contains dedicated disk storage. The servers' participants in the cluster are interconnected via high-speed network.

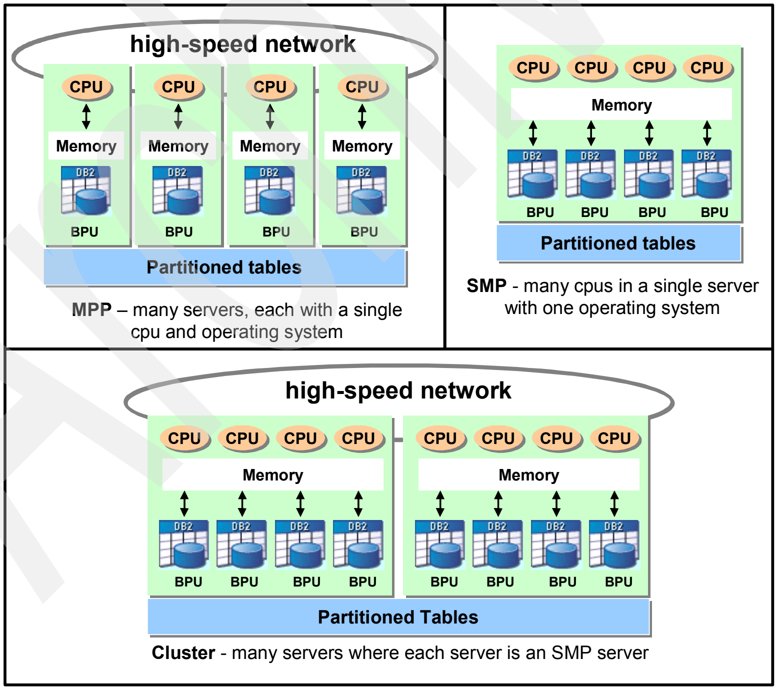


Figure 5-13 DB2 database topology

A shared nothing database architecture can be hosted on a single SMP server, where the partitions of the database would all reside on the server. The shared nothing database can be used in an SMP environment. An SMP's processing capability can be increased by adding processors, memory, and disk to it. This is referred to as *vertical* or *scale up* of the database.

However, it is also possible to spread the database partitions over a number of servers to form a cluster of servers that are connected by a high-speed, scalable communication network. This allows the database to scale in a linear fashion beyond the limits of a single physical server. This is referred to as *horizontal* or *scale out* of the database.

5.5.4 Balanced Configuration Unit (BCU)

The principle of the balanced configuration unit is to balance a defined combination of resources related to the data warehouse. These resources include processors, memory, I/O, storage, DB2 database partitions (BPUs), and DB2 configuration parameters combined under a single operating system.

The resources are divided and balanced to comprise a single practical building block that is scalable. Larger systems can be configured by combining several building blocks into one system image. The name given to this basic building block is the *balanced configuration unit*, or BCU.

The BCU is the minimum replicable hardware and software stack necessary to start or expand the infrastructure of the BI system and provides a scalable performance ratio of disk I/O to memory to CPU to network.

A balanced configuration avoids bottlenecks that can limit overall performance. Balancing also reduces the risk of oversizing single components.

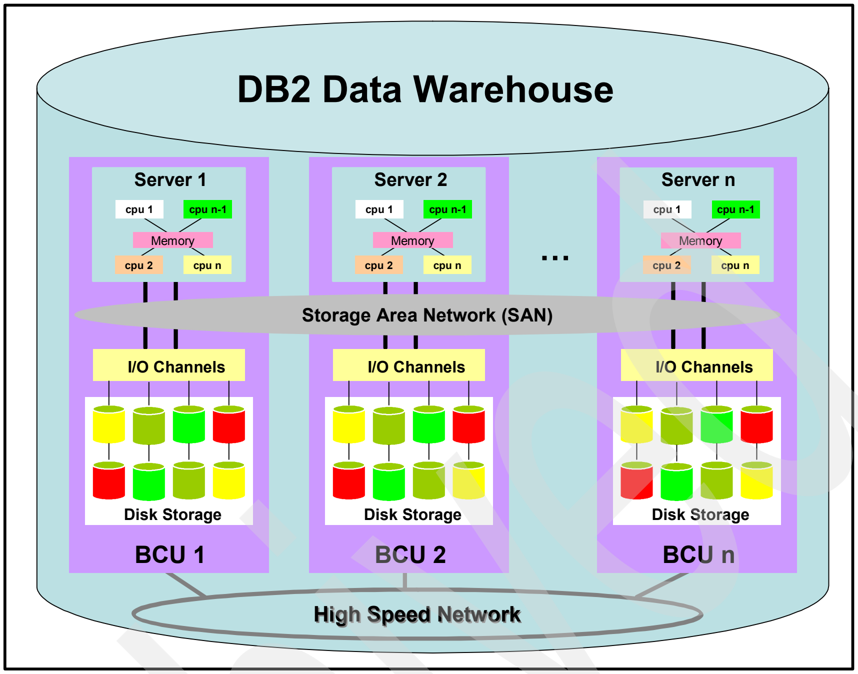


Figure 5-14 BCU: Balanced Configuration Unit

The goal of the BCU is to provide a prescriptive and quality approach through the use of a proven balanced methodology. By using the BCU concept when implementing a data warehouse, you can reduce your total time to implement and lower the total cost of ownership (TCO) of the data warehouse. The prescriptive approach used by the BCU minimizes the complexity of data warehouse design and implementation via standardized and tested designs and practices that increase the quality and manageability of the data warehouse.

The BCU provides many benefits, such as:

- Taking a complex concept and reducing it to more easily understood units.
- Scaling the data warehouse is simplified. As business requirements change and more data sources are identified, BCUs can easily grow the data warehouse to meet new workload demands.
- Over time, workloads deployed on consistent BCU configurations lead to improved sizing and capacity-planning processes for the data warehouse.
- The BCU provides a prescriptive approach to implementing the total IBM BI solution. The prescriptive approach is intended to reduce the risk of sizing, deployment, integration, and planning for growth. In addition, the prescriptive

approach provides best practice knowledge and intellectual capital because by defining building blocks, performance, scalability, and reliability can be better understood.

- A consistent approach and configuration allow more focused quality testing therefore reducing the number of variables and reducing risk.

The BCU provides the foundation for the end goal of more reliable and stable BI solutions. The following aspects of the BCU help to achieve that goal:

- Detailed implementation specification, including storage structures such as table spaces.
- Because the BCU is a building block for constructing a data warehouse, understanding a single BCU leads to understanding the entire infrastructure.
- IBM BCU solution offerings are quality-tested and validated.
- As a result of months of testing, repeatable methodologies and best practices are gathered and documented.

5.5.5 DB2 delivers performance for BI

Databases today increasingly combine traditional online access with reporting and decision support infrastructures. In such mixed environments, it is extremely common to have queries of differing complexities. There are queries considered of short duration that generally require few resources and are expected to execute very quickly. But there are also queries considered of long duration that in general consume significant computing resources. In these environments, it is important to prevent reporting queries from monopolizing all the system resources so that online operations can continue to execute quickly.

DB2 Query Patroller is a tool that enables improved workload management and data warehouse administration. It can help address the needs of both users and DBAs. It maximizes system resources by:

- Intercepting runaway queries before they can degrade performance
- Running short or canned queries with more consistent response times (long, complex statements can be queued, blocked, or scheduled to run during off-peak hours)
- Prioritizing the most urgent queries and important users or groups
- Preventing users or departments from monopolizing the enterprise data warehouse computing resources
- Preventing overload of the database computing resources during peak workload times

- Reducing the impact of expensive queries by running them at scheduled off-peak times

DB2 Query Patroller intercepts and evaluates queries to determine if they should be run immediately or if they should be queued or blocked for later execution. It also allows DBAs to assign priorities to queries run by different users or groups and can prioritize based on the class of query being executed.

DB2 Query Patroller lets administrators set database thresholds for the:

- Number of queries users can execute concurrently.
- Maximum cost of a query users can run.
- Total cost of all concurrently running queries.
- Total number of queries (of any size) that can execute concurrently.
- Total number of queries of a particular size that can run concurrently.

Using the capabilities of DB2 can insure that your BI queries provide the performance you need to satisfy your users.

6

Case study software components

Business innovation and optimization (BIO) is driven through a combination of event, process, and information services that:

- Supports continuous awareness and improvement
- Enables faster and more effective decisions
- Facilitates the attainment of business objectives
- Helps manage operations efficiently and effectively

In this chapter, we present and describe the product components used in the redbook case study outlined in Chapter 7, "Performance insight case study overview" on page 247 and implemented in Chapter 8, "Performance insight case study implementation" on page 259. We have included an expanded and detailed set of component descriptions here for your benefit to minimize the requirement of accessing other documentation. There are a significant number of components, and it is important that their framework and integration are well understood. Particularly, because this is a relatively new initiative and requirements are frequently updated.

To aid in understanding, we have included additional information to create a *one-stop-location* for access to this integrated type of information. We have included product details and functionality. But, in addition, there is information describing the usability, positioning, and some implementation recommendations for the products. We think it can be of significant benefit for your overall perspective.

The case study is based on the Performance Insight On-Ramp of the IBM Business Innovation and Optimization initiative. We developed and deployed the case study, described in Chapter 8, “Performance insight case study implementation” on page 259, using the following products:

- DB2 Data Warehouse Edition V9.1 (DWE)
- WebSphere Information Integration
  - WebSphere Information Integrator V8.3
  - WebSphere Data Stage V8
- WebSphere Portal Server V5.1
- WebSphere Business Modeler V6.0.1
- WebSphere Business Monitor V6.0.1
- WebSphere Integration Developer V6 and WebSphere Process Server V6
- WebSphere Message Broker V6 and Advanced Enterprise Service Bus

The following sections provide detailed descriptions of those products.

6.1 DB2 Data Warehouse Edition

Business intelligence (BI) is a critical business advantage. Most businesses have some level of BI capability, but they must continue to grow their capabilities to remain competitive. DB2 Data Warehouse Edition is a powerful set of products that rapidly enables enterprises to develop robust BI solutions, such as those with data mining techniques, inline analytics, and an integrated design tool.

The integration of BI and business process management can take you to a new level of competitive advantage. This advantage is created because typical BI solutions do not embrace processes, and that is required to get to the next level of advantage. It is crucial to better understanding of the company and the optimization of the business processes. It is this integration that gives the business processes access to all the enterprise consolidated information on a granular level. That information can enable improved problem recognition and resolution, or, better yet, problem avoidance. It is this integrated information that can support operational, tactical, and strategic decision making to enable management to meet their business performance measurements and business goals.

The integration of BI and business process management is what we refer to as Performance Insight. That is, it delivers actionable insights throughout the enterprise to meet those business measurements and goals. DWE provides the products to enable the solution.

DB2 Data Warehouse Edition (DWE) V9.1 is an integrated platform for developing data warehouse-based analytics, including Web-based applications with embedded data mining and multidimensional Online Analytical Processing (OLAP). DWE integrates core components for data warehouse administration, data mining, OLAP, and inline analytics and reporting. These platform pillars are based on the technology of Rational Data Architect together with the SQL Warehousing Tool, DWE Mining, DWE OLAP (based on DB2 Cube Views), and DB2 Alphablox.

In DWE Design Studio, physical data modeling, cube modeling, data mining modeling, and SQL data flow/control modeling are unified in one common design environment. That Eclipse-based design environment integrates all of the DWE products within a common framework and user interface (the one exception is DB2 Alphablox, which currently uses a native interface. However, the direction is to support the Eclipse plug-in architecture in a future release).

DWE is a component-based architecture with client and server functions, both leveraging emerging IBM Software Group frameworks and shared subcomponents. DB2 is the foundation for DWE, providing a scalable data warehousing platform. DWE then extends the DB2 data warehouse with analytic tooling (design-side) and infrastructure (runtime), including WebSphere Application Server and Rational Data Architect. DB2 Alphablox is the tool for developing custom applications with an embedded analytics-based dashboard. DWE gives customers faster time-to-value for enterprise analytics, while limiting the number of vendors, tools, skill-sets, and licenses required.

DWE OLAP works together with Alphablox to accelerate OLAP queries. It uses multidimensional models to design runtime DB2 objects containing critical dimensions and levels, or slices, of the cube. These pre-joined and pre-aggregated Materialized Query Tables are exploited by the DB2 optimizer, which rewrites incoming queries and transparently routes eligible queries to the appropriate MQT for significantly faster query performance.

Besides performance benefits, DB2 OLAP metadata and tooling allow cube models to be defined once in DB2 and also used by Alphablox. Either way, the productivity of OLAP administrators and developers is improved. Because the shared common metadata includes aggregation formulas and calculations, users benefit from greater consistency of analytical results across the enterprise.

Embedded data mining capability in DWE uses IBM DWE Mining algorithms to analyze data warehouse data, in place, and provide insights into business behaviors that are otherwise unknown, invisible, and impossible to discover. Data mining in the data warehouse enables improved accuracy and timeliness. In the past, data mining was invoked by an SQL programming interface. In DWE Design Studio, a new data discovery function allows you to profile the data, sample and view table contents, and visualize correlated statistics to understand which parts of the data warehouse hold the best potential for data mining. Next, the Eclipse data flow editor is used to visually design data mining flows with modeling, scoring, and visualization operators. Then, SQL can be generated and pasted into an Alphablox page, or any customer application, to invoke the data mining flow for embedded analytics.

DB2 Alphablox lets you quickly and easily build custom Web-based OLAP-style reporting either as stand-alone applications or embedded in Web portals, dashboards, and other existing applications. Time-to-value comes from leveraging prebuilt libraries of J2EE components, the *building blox* of this architecture. Because of the focus on customization and the capability to embed, Alphablox enables applications with embedded analytics to invoke the data warehouse-based analytic structures (data mining and OLAP) modeled and maintained in DWE. We describe DB2 Alphablox in more detail in 6.1.1, “DB2 Alphablox” on page 128.

The new SQL Warehousing Tool generates DB2-optimized SQL based on a visual data flow modeled in the Design Studio canvas, drawing from a palette of predefined operators. The library of SQL operators covers the types of transformations needed to populate analytic structures involved in data mining and OLAP, or for any in-database data flows. Data flows can be combined in sequences as control flows which are then scheduled for execution. Because the data flows are SQL-based, DB2 acts as the runtime engine, with WebSphere Application Server providing control and scheduling.

In summary, the new Design Studio in DWE V9.1 provides an integrated platform for modeling, designing, and maintaining data warehouse-based analytic structures, which can be invoked by Alphablox for true embedded enterprise analytics.

DB2 DWE V9.1 is available in two versions:

- DB2 Data Warehouse Base Edition
- DB2 Data Warehouse Enterprise Edition

Table 6-1 on page 127 provides a brief comparison of the components in the two DB2 Data Warehouse Editions.

Table 6-1 DB2 Data Warehouse Edition components

Base	Enterprise	Product
Yes	Yes	DB2 UDB Enterprise Server Edition
Yes	Yes	DWE OLAP
Yes	Yes	DWE Design Studio
Yes	Yes	DWE Integrated Installer
No	Yes	DB2 Data Partitioning Feature
No	Yes	DB2 Query Patroller
No	Yes	DWE Mining
No	Yes	DB2 Alphablox
No	Yes	DWE Admin Console
No	Yes	DB2 Design Studio
No	Yes	DWE SQL Warehousing Tool

DB2 UDB ESE is the most comprehensive edition and is designed to meet the relational database server needs of mid-size to large-size businesses. It can be deployed on Linux, UNIX, or Windows servers of any size, from one CPU to hundreds of CPUs. DB2 UDB ESE is an ideal foundation for building enterprise-wide solutions for On Demand Business, such as large data warehouses of multiple terabyte capacity, or high performing 24x7 availability

high volume transaction processing business, or Web-based, solutions. It is the database edition of choice for industry-leading enterprise solutions.

DB2 UDB ESE is available on all supported versions of UNIX (AIX, Solaris, and HP-UX (including HP-IA64), Linux, Windows NT® (SP6 or later), Windows Server 2000 (SP2 or later), and Windows Server 2003. It does not run on Windows XP for production purposes, but can be licensed for user acceptance testing, test, and application development on this platform (this restriction is in accordance with the Microsoft direction for this operating system and therefore applications running on Windows XP can be adequately serviced by DB2 Express, DB2 WSE, or DB2 WSUE servers).

Connectivity to zSeries-based and iSeries-based data is provided by the DB2 Connect component and is ideal for certain data replication scenarios and remote administration.

6.1.1 DB2 Alphablox

DB2 Alphablox has a special role in BIO, which is delivering data cubing services for development and support of information dashboards and interactive data analysis. By *data cubing*, we mean the construction of multidimensional data cubes for enabling multidimensional analysis.

DB2 Alphablox and all DB2 Alphablox analytic-enabled solutions run as J2EE-compliant applications in an application server, and they are accessed by using a Web browser. Unlike traditional query and reporting tools that interact with application servers, DB2 Alphablox leverages the application services, portal services, and integration broker services provided by the application server. In addition, DB2 Alphablox leverages the common foundation for developing, deploying, and maintaining distributed applications.

DB2 Alphablox architecture

DB2 Alphablox is comprised of the following elements:

- Platform
- Analytic-enabled solutions
- Administration application
- Application server adapters

The platform, the core component of DB2 Alphablox, runs within the business tier of the J2EE application server. While running as a J2EE application within the host application server, it also provides the services of a fully functional analysis server. For DB2 Alphablox analytic-enabled solutions to fully leverage the analytic capabilities and services of DB2 Alphablox, the platform requires a separate installation for components and adapters that are not traditionally part of J2EE applications. Figure 6-1 on page 129 depicts these components.

The adapters allow DB2 Alphablox to communicate with each supported application server to perform administration functions. Many of these functions, such as defining applications, are set up differently on each application server.

DB2 Alphablox analytic-enabled applications run as application server applications within the Web tier. The applications, while interacting with DB2 Alphablox, are configured as separate and encapsulated J2EE applications. Updates to DB2 Alphablox-enabled applications can be deployed, backed up, upgraded, and migrated independently of the DB2 Alphablox platform.

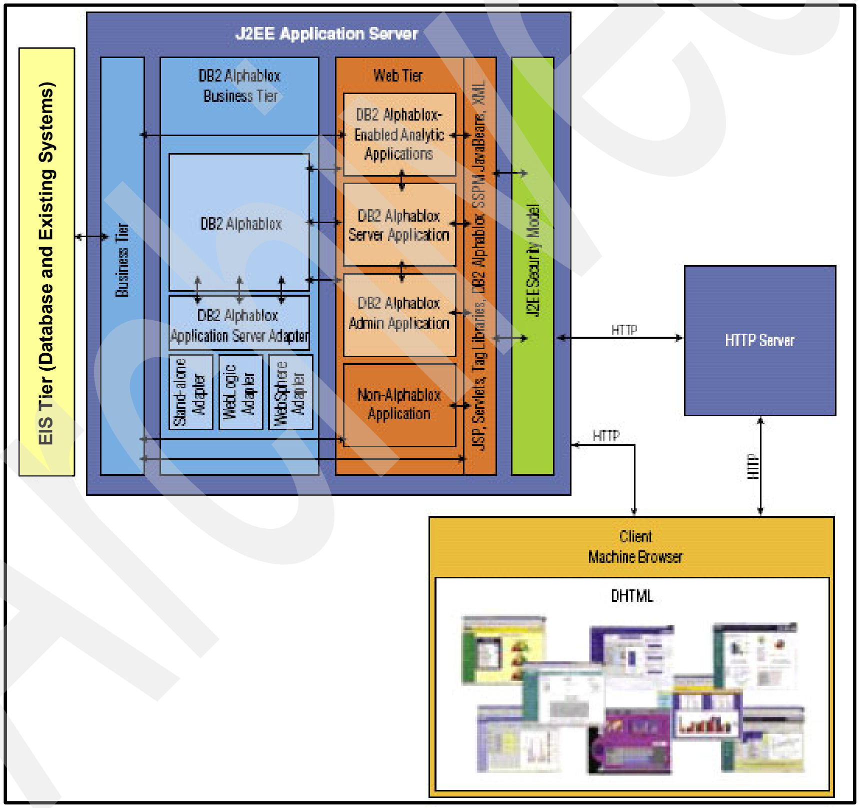


Figure 6-1 DB2 Alphablox architecture

DB2 Alphablox also registers two J2EE applications within the Web tier of the application server. They are the DB2 Alphablox server application and the DB2

Alphablox administration application. The application server manages DB2 Alphablox in the same way it manages any other Web application. For example, it is auto-started by invoking a servlet. DB2 Alphablox is then suspended and resumed by the application server as needed, based on requests received by the application server and the management model.

DB2 Alphablox analytic components

DB2 Alphablox enables organizations to integrate analytics across functions and lines of business and to deploy analytic solutions for improved decision making. The technology enables organizations to optimize various aspects of their business, including:

- Self-service reporting and analysis applications
- Operational analysis applications
- Financial reporting and analysis applications
- Planning applications
- Business performance and key performance indicators (KPIs) for interactive information dashboards

Data can be presented in several formats, including:

- Interactive grids, charts, and reports
- Informational dashboards
- Planning and modeling applications
- Information portals

DB2 Alphablox can integrate data from all enterprise information resources, including relational and multidimensional databases, transaction systems, and other outside content feeds. This ensures that users have immediate access to all pertinent data, regardless of where or how it is stored. In addition, users can utilize a write-back capability to facilitate real-time planning and modeling applications.

DB2 Alphablox applications

DB2 Alphablox-enabled applications have the following characteristics, that can be implemented using various combinations of DB2 Alphablox features:

- Interactive and guided analysis
- Real-time data access, analysis and alerts
- Personalization
- Sharing and collaboration
- Real-time planning through write-back

Interactive and guided analysis

DB2 Alphablox-enabled applications enable users to interact with real-time data via grids and charts, as well as other components, such as drop-down lists.

These interactive analytic components are served in dynamic HTML, based on Dynamic HTML (DHTML) technology, utilizing JavaScript and cascading style sheets (CSS). The DB2 Alphablox Dynamic HTML client provides the benefits of easy deployment with interaction.

For example, a user can interact with a grid and have just that grid updated rather than having to refresh the entire page.

Users perform multidimensional analysis by manipulating the data displayed in the grid and chart, as Figure 6-2 depicts. Analysis actions, such as drilling, pivoting, sorting, and selecting can be performed directly on the grid and chart, through toolbar buttons, through right-click menu options, or via the DB2 Alphablox form-based controls and components added by application developers.

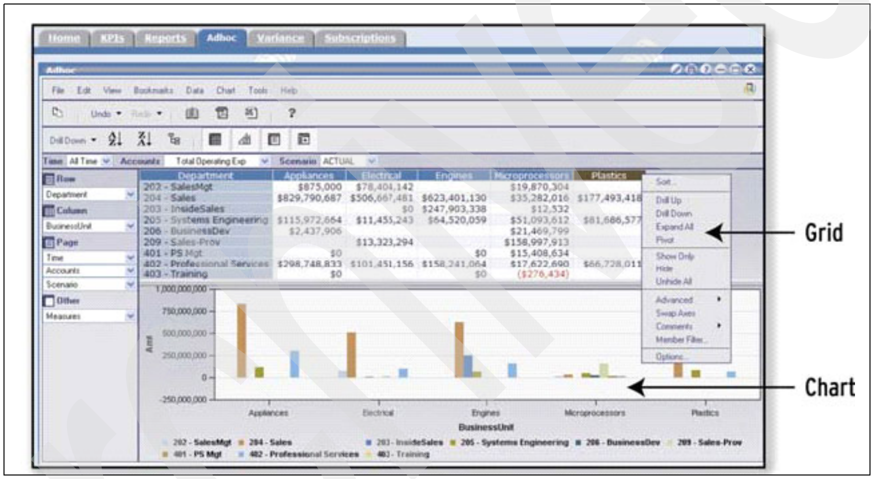


Figure 6-2 DB2 Alphablox Grid and Chart

Real-time data access and analysis

DB2 Alphablox-enabled applications can drive analysis of data from multiple data sources, both relational and multidimensional, including DB2 Cube View cubes. Through the native ability to query a database, DB2 Alphablox exposes the analytic functionality in the database engine. Users can leverage capabilities such as ranking, derived calculations, ordering, filtering, percentiles, variances, standard deviations, correlations, trending, statistical functions, and other sophisticated calculations while performing analysis.

For example, a controller of a manufacturing company could choose to look at key performance indicators (KPIs) such as profit, bookings, billings, backlogs, trends, and comparisons of actuals to budget, as Figure 6-3 on page 132 depicts. The data is real time and the controller can choose to drill down on various items, such as total revenue, to get more detail.

Figure 6-3   DB2 Alphablox example: Comparisons of actual to budget

The DHTML client in DB2 Alphablox is very flexible. Data can be presented the way users need to see it. For example in Figure 6-4, a controller wanted to see a butterfly report in which the prior three months of actual figures are shown to the left of the Accounts row headers and the current month details are shown to the right of those headers.

Comp Time: Last 3 Months
Business Unit: Total Corporation
Department: All Departments
Base Time: Current Month

Feb	Mar	Apr		May			
ACTUAL	ACTUAL	ACTUAL	Accounts	ACTUAL	BUDGET	Variance	Varia
275,522 K	202,728 K	195,090 K	**Payroll & Benefits**	230,320 K	298,227 K	67,907 K	
324 K	1,125 K	503 K	**Employee Devel**	862 K	3,237 K	2,375 K	
73 K	0 K	2 K	**Recruiting**		2,314 K	2,314 K	
491 K	195 K	197 K	**Admin Fees**	102 K	265 K	163 K	
23,284 K	11,737 K	(4,285 K)	**Professional Fees**	6,185 K	6,227 K	42 K	
33,492 K	25,001 K	20,038 K	**T & E**	21,143 K	20,022 K	(1,121 K)	
6,290 K	8,242 K	10,954 K	**Marketing**	5,428 K	9,921 K	4,493 K	
539 K	196 K	665 K	**Events**	187 K	971 K	783 K	
5,176 K	1,637 K	1,186 K	**Equipment Cost**	1,644 K	4,053 K	2,409 K	
15,109 K	14,974 K	16,754 K	**Depreciation Exp**	14,009 K	16,913 K	2,904 K	
22,311 K	71,743 K	39,362 K	**Occupancy Exp**	28,629 K	40,886 K	12,258 K	
10,898 K	7,250 K	7,120 K	**Office Exp**	8,089 K	12,667 K	4,577 K	
(33,506 K)	(55,078 K)	(48,367 K)	**Allocations**	(37,837 K)	(57,547 K)	(19,710 K)	
31 K	83 K		**Other**				

Figure 6-4 DB2 Alphablox butterfly report

Personalization

Users have different data and business needs. Therefore, DB2 Alphablox analytic-enabled solutions can be personalized to meet the needs of each user. For example, the first logon window that users see can be customized according to their role in the organization. Users in the sales department can see the top five best-selling products or the most profitable regions for month-to-date. Users in finance might be more interested in monthly summary figures for sales, cost of goods, marketing, payroll, and profit, as shown in Figure 6-5.

Figure 6-5 DB2 Alphablox customization example

In addition, each DB2 Alphablox analytic-enabled solution can contain custom user preference windows that enable users to personalize the solution to their needs, as Figure 6-6 on page 134 depicts. In this example, the user can choose the business units and accounts that are displayed in the dial gauges.

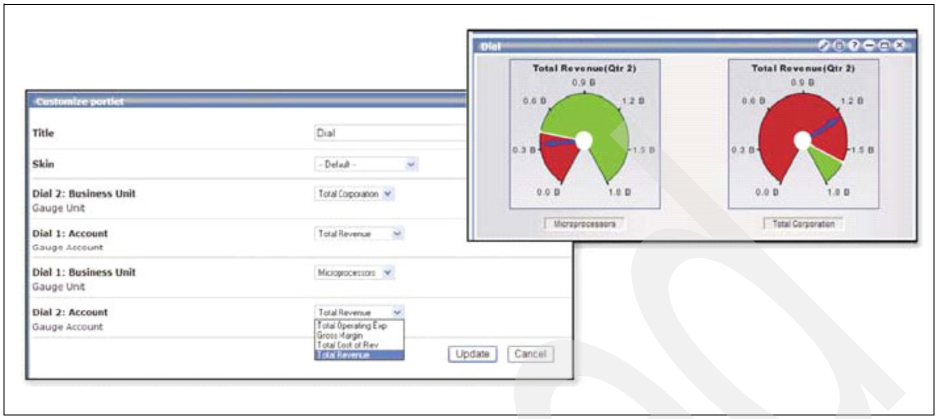


Figure 6-6 DB2 Alphablox personalization example

Sharing and collaboration

DB2 Alphablox analytic-enabled solutions support collaboration, enabling users to leverage existing messaging, and workflow systems to save and share application views once the analysis is performed. In addition, DB2 Alphablox supports collaboration features such as bookmarking, e-mail, and PDF generation.

Real-time planning through write-back

Analytic applications can range from historical analysis to forward-looking forecasting and proactive resource allocation. The DB2 Alphablox data *write-back* capability enables developers to build real-time planning applications, such as budgeting, sales forecasting, *what-if* modeling, and collaborative demand planning, as you can see in Figure 6-7.

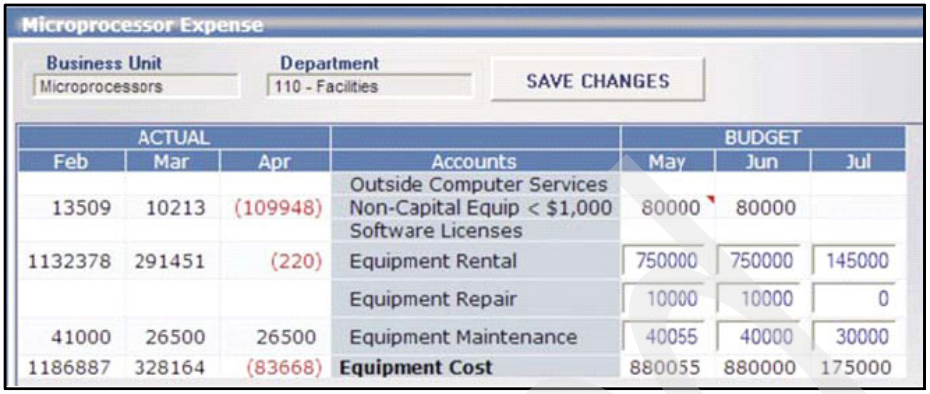


Figure 6-7 DB2 Alphablox what-if modeling example

DB2 Alphablox and the application server

Enterprises gain competitive advantage by quickly developing and deploying custom applications that provide unique business value.

The J2EE standard provides an opportunity for analytic solutions to undergo a true paradigm shift. Prior to J2EE, there was not a standard, cross-platform architecture that would enable truly distributed computing in a Web environment. J2EE simplifies enterprise application development and deployment in several ways:

- Development environment based on standardized, modular components
- A complete set of services to application components
- The ability to extend existing services and add new services that provide complete interoperability with standard services
- The capability to handle the details of application behavior without complex programming

The DB2 Alphablox architecture capitalizes on this standard, cross-platform environment to deliver analytic solutions. DB2 Alphablox draws on Java technologies to implement a Web-based, N-tier architecture for delivery of analytic solutions. J2EE provides the framework for distributed, multi-tiered applications. Application logic is divided into components according to function. The most common configuration is a three-tier configuration which Figure 6-8 on page 136 depicts, which consists of the following:

- The Enterprise Information Systems (EIS) tier, also known as the Database tier, runs on database servers. Data resides on these servers and is retrieved from relational and multidimensional data servers.

- The J2EE application server is host to the business and the Web tiers. The business tier is the code that implements the functionality of an application, and the Web tier supports client services through Web containers.
- The Client tier is where client applications are presented. For a Web-based J2EE application such as DB2 Alphablox, the Web page display and user interface presentation occur on a client machine through a Web browser. The Web browser downloads Web pages and applets to the client machine.

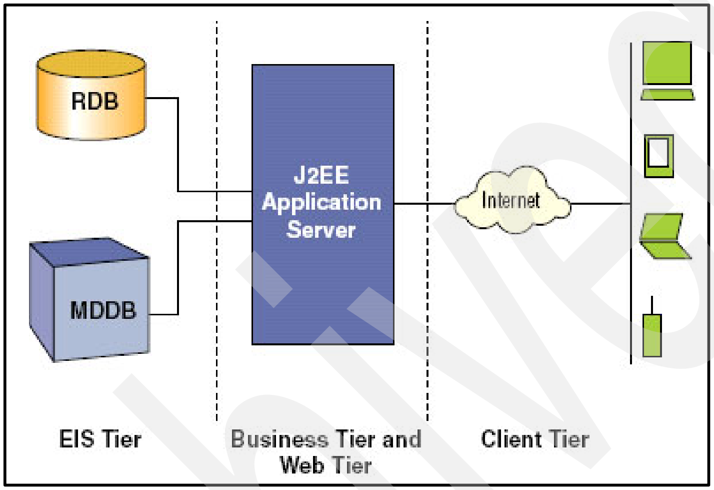


Figure 6-8 Three-tier configuration

Within the J2EE framework, DB2 Alphablox runs as an application within the application server, as you can see in Figure 6-9, leveraging existing server resources, such as process management, application management, and request management. DB2 Alphablox-enabled applications run as standard application server applications and the Web pages are served directly through the application server.

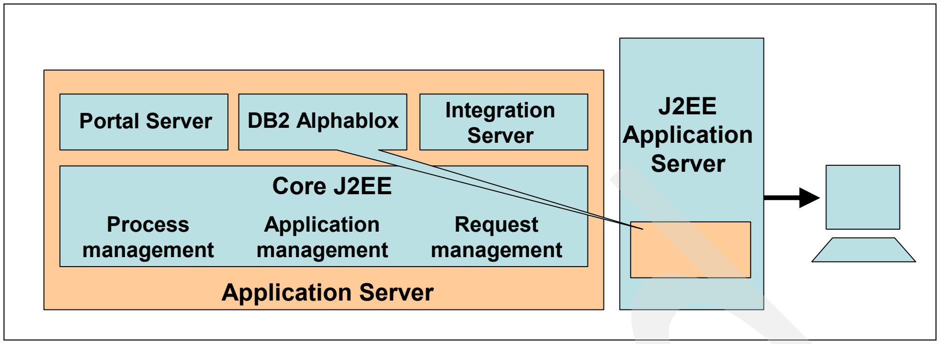


Figure 6-9 DB2 Alphablox running on an application server

Components of a DB2 Alphablox-enabled application

Once installed, DB2 Alphablox provides a comprehensive set of components and application templates for developing analytic solutions. The modular *building Blox* approach enables fast delivery of personalized and customized applications.

DB2 Alphablox analytic-enabled applications appear as a collection of Web pages that serve as containers for the following application components:

- Standard HTML tags and page elements (logos, text, or images) to enhance the user interface
- Blox necessary to deliver the required application functionality
- JavaScript for extended application and user interface (UI) logic (optional)
- Java servlets for customized business logic (optional)

Application building Blox

To promote the creation of custom analytic solutions, DB2 Alphablox includes a set of generic application building Blox, as Figure 6-10 depicts. Application building Blox are prebuilt, high-level JavaBean components that provide the functionality required by analytical applications. Blox allow developers to perform data manipulation and presentation tasks and build dynamic, personalized analytic applications. Because Blox are modular and reusable in design, they are easily built into a variety of analytic solutions.

Each Blox provides broad functionality through its properties and associated methods, which allow the Blox appearance and behavior to be specified and controlled. Event filters and event listeners are available for performing pre-event and post-event processing for user events such as drilling up or drilling down, pivoting, changing the page filter, loading a bookmark, or changing the data value in a grid cell.

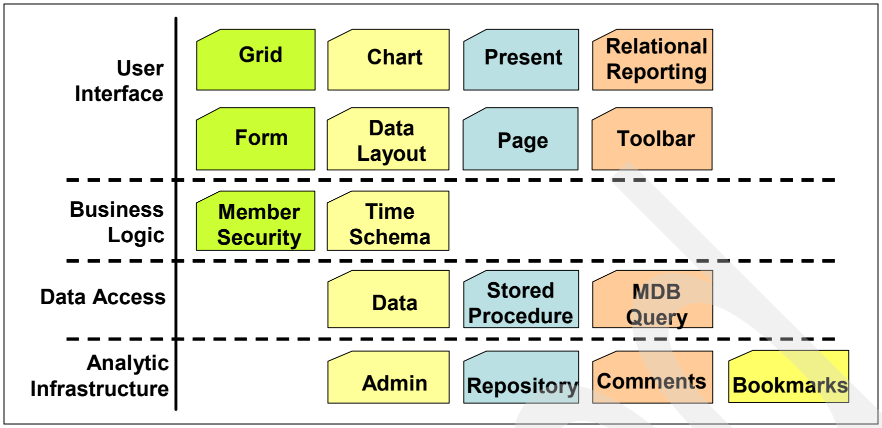


Figure 6-10 DB2 Alphablox application building blox components

Deploying DB2 Alphablox

Detailed descriptions of the aspects for deploying DB2 Alphablox-enabled applications are described in the following sections.

Administration

DB2 Alphablox is designed for integration with an existing application server environment to help leverage a robust architecture to deliver analytic solutions. To facilitate installation, DB2 Alphablox provides a cross-platform, graphical interface to install the DB2 Alphablox server and administration pages. Once installed, you have centralized administration of analytic solutions through a set of administration pages. These pages enable application developers to manage DB2 Alphablox services that use the same resources and are also complementary to the administration provided through the application server.

For example, application developers use the DB2 Alphablox administration pages as a convenient way to register and set up new applications. When creating an application from the DB2 Alphablox home page, DB2 Alphablox creates the application definition in the DB2 Alphablox repository. In addition, the J2EE context and directory structure are created.

When running DB2 Alphablox analytic-enabled solutions, the application server passes the user name and associated roles to DB2 Alphablox. To allow personalization, the user profile can be configured to allow the application developer to define custom properties for the user.

DB2 Alphablox administration pages also can be used to configure DB2 Alphablox specific settings such as data sources, relational cubes, groups, and DB2 Alphablox server settings. These administration pages, packaged as a

J2EE-compliant application, can be managed through the same mechanisms as any other Web application in the environment.

DB2 Alphablox can be administered either through Web pages under the DB2 Alphablox home page or through a standard command console accessible through any Telnet terminal software. Using either method, administrators can create users, data sources, and other DB2 Alphablox objects. This design enables remote server administration. For example, an application developer can define a new application from a workstation using the Web pages, or an administrator can perform routine monitoring tasks on a remote computer.

Setting up the application

Once the DB2 Alphablox-enabled application is completed, it is a self-contained J2EE Web application that authorized users can access as they would any other Web page. The application developer defines the DB2 Alphablox-enabled analytic application through the appropriate DB2 Alphablox administration page. The application developer specifies information such as application context, display name, home URL, and default saved state. Based on this information, DB2 Alphablox creates the application definition in the DB2 Alphablox repository and the application is made available to users through the application server.

The application context is a J2EE term referring to the descriptor that uniquely identifies the Web application or module. The application context serves as a container for J2EE applications that run within a J2EE server. Because they are standard J2EE applications, it is easy to package them as Web archive (WAR) or enterprise archive (EAR) files so they can be deployed to various application servers.

As specified in the J2EE standard, each DB2 Alphablox-enabled analytic application has a WEB-INF directory that houses the configuration information and the supporting resources necessary to deploy the application. These resources include components such as Java classes (Java archive files) and JSP tag libraries.

The WEB-INF directory also includes the Web application descriptor file web.xml. The web.xml file, a standard file in all J2EE applications, is an XML file that contains markups that define the application behavior internally and as it relates to the application server. Included in the web.xml file are application-specific properties and their values, servlet mappings, and security constraint information. This file enables the deployment into application servers, because it includes everything the application server needs to know. The application server reads the web.xml file during initialization of the application.

Managing metadata in the DB2 Alphablox repository

The Metadata Repository Manager controls the contents of the DB2 Alphablox repository. The repository is a database that holds application-specific metadata for applications and users. It also includes information on data sources, relational cubes, user groups, roles, applications, and application states. When a user saves an application or Blox state, it is stored in the repository. The repository also stores bookmarked Blox properties that enable collaboration between users and groups, as well as XML representations of saved spreadsheet Blox.

DB2 Alphablox at runtime

To support a widely dispersed user community, DB2 Alphablox provides a high degree of flexibility by allowing the developer or user to choose the delivery format of DB2 Alphablox-enabled applications at runtime. The same application can be deployed in different modes at different times to meet different requirements throughout the enterprise. This arrangement enables all users to leverage analytic solutions, regardless of any network bandwidth or client-side software limitations. It also allows applications to be optimized according to the application function and analytic capability required by the user.

DB2 Alphablox application deployment options

Consider the following deployment scenarios, which Figure 6-11depicts:

- Static HTML: The application is delivered over an extranet or a narrowband network, providing users with simple data views. No significant client processing is required, and the information is presented in static HTML.
- Dynamic HTML: This mode utilizes JavaScript and cascading style sheet (CSS) to support the full range of data analysis functionality with a highly usable and customizable graphical user interface. It does not require any plug-ins or downloads of Java class files.
- XML rendering: The application data needs to be integrated with transactional application servers or delivered to clients such as cell phones or pagers. The application is rendered in XML for delivery to XML-enabled applications and clients.
- Ready for print: Users can request that pages be rendered for printing. The application presents the information, formatted for printing.
- Ready for PDF: The application user requires a report with greater control over page layouts, storage, and printing. The application view is converted to PDF.
- Export to Excel or spreadsheet Blox: The application provides data to be analyzed in Excel or spreadsheet Blox and exports the data to the chosen application.

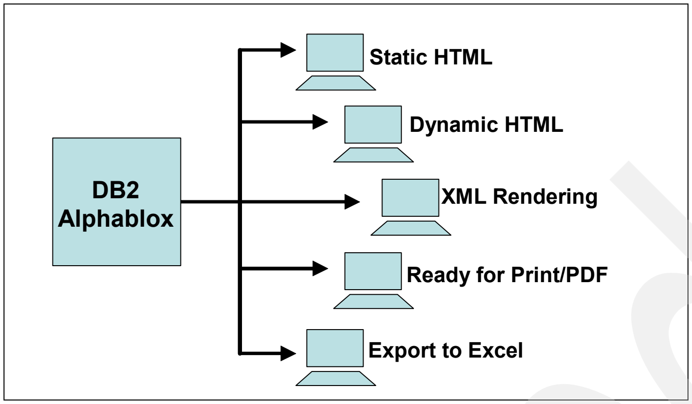


Figure 6-11 DB2 Alphablox scenarios

DB2 Alphablox services

In addition to application building Blox, the DB2 Alphablox platform consists of several services that help manage the applications, as you can see in Figure 6-12. Each DB2 Alphablox service is responsible for a particular aspect of the application operating environment.

In this section, we take a closer look at each of these services.

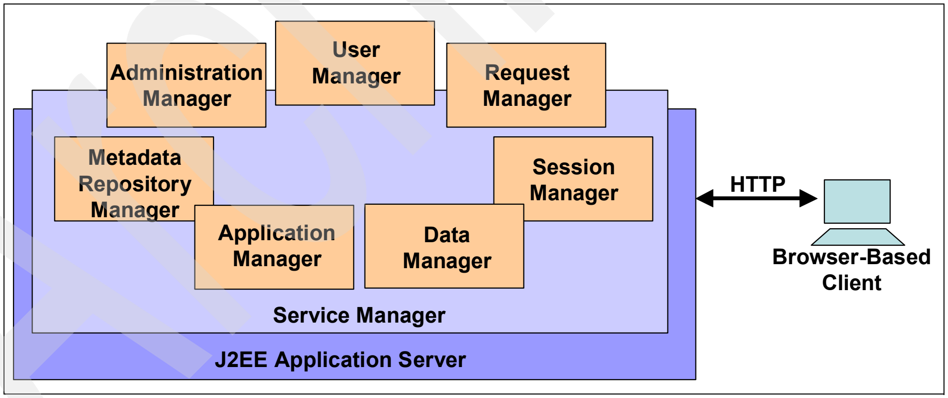


Figure 6-12 DB2 Alphablox services

Service Manager

As the focal point for server administration and monitoring, the Service Manager starts, stops, and provides access to the other managers, passes service requests to the correct manager, and monitors DB2 Alphablox resources.

Request Manager

The application server processes the original HTTP request; if there is DB2 Alphablox content, it is passed to the Request Manager for processing. If the request is from a user for whom there is no active session, the Request Manager passes the request to the Session Manager. The Request Manager processes the application and Blox names and then passes this information to the Application Manager for further processing.

As the application runs, the Request Manager coordinates communications between Blox on the application page and their server-side peers. The Request Manager also creates, monitors, and manages threads for each request.

Session Manager

The Session Manager establishes a session for each new DB2 Alphablox browser instance. If an individual user has multiple DB2 Alphablox browsers open, the user would have multiple concurrent sessions. The Session Manager creates and manages session objects and tracks which applications a user visits. It also maintains a mapping between the DB2 Alphablox session and the session being maintained by the application server. The Session Manager also terminates dormant sessions after first saving the current state of each application and releases session resources.

User Manager

The application server passes the user name to the User Manager, which gets the user information from the request object and then interacts with the application server through standard APIs to ensure that the user is authenticated. It controls all users of DB2 Alphablox services and creates and manages user instances. The User Manager also monitors resources allocated to each user and maintains user information, such as which applications are accessed, by which users, and for how long.

The DB2 Alphablox User Manager manages user authentication and authorization as well as provides personalization capabilities for customizing application content. By default, DB2 Alphablox uses its repository and the J2EE Security API to manage user and group information.

DB2 Alphablox also provides an out-of-the-box Lightweight Directory Access Protocol (LDAP) integration solution. This solution allows DB2 Alphablox to authenticate and authorize the users by using an LDAP directory server to recognize DB2 Alphablox users, groups, and custom properties. The DB2 Alphablox User Manager is built on top of the personalization engine called the *Extensible User Manager*. For environments in which custom security is necessary, the Extensible User Manager personalization engine provides interfaces that allow the extension of either of the out-of-the-box security solutions, DB2 Alphablox repository or LDAP. It is also possible to plug-in another external user manager, such as NTLM or some existing Enterprise JavaBeans (EJBs).

Application Manager

The Application Manager is responsible for creating or modifying the DB2 Alphablox application definition from the DB2 Alphablox administration applications page. The Application Manager verifies user privileges for application access, starts application instances, manages each application instance, and supervises page processing before a page is presented to the user. The application design determines the exact page processing that occurs.

Application instance

The application instance controls the running state of each application. There is an application instance for each DB2 Alphablox browser instance in which the application is running. It is important to understand the difference between an application and an application instance, which Figure 6-13 on page 143 depicts.

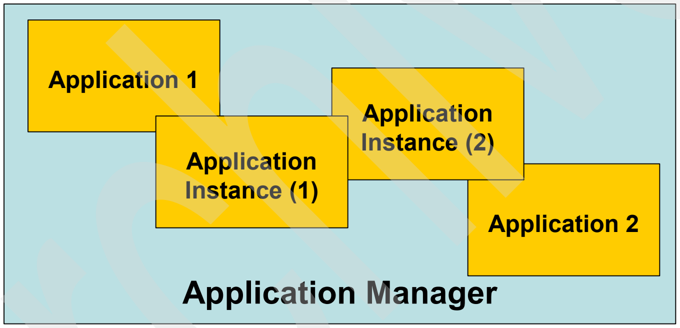


Figure 6-13 Application services

An application is the entity JSP, HTML pages, images, servlets, and so on that the application developer creates and stores on the DB2 Alphablox server. An application instance is the running state of that application, appearing in a browser window and interacting with a single user. The instance remains active until the client or administrator stops the application or the session times out.

The application instance maintains information about the state of each Blox in the application as well as information about the application as a whole. A user can choose to save the state of the entire application or simply that of an individual Blox. This feature can enhance collaborative analysis by enabling users to return to the saved state of the application and to share their results with others.

Data Manager

The Data Manager controls application requests for data sources, and is responsible for accessing, browsing, querying, and retrieving data from data sources as Figure 6-14 on page 144 depicts. It is uniquely designed for connectivity to a variety of data sources. Through a single API for both multidimensional and relational sources, the Data Manager translates the data into dimensions, rows, columns, and pages, the components used in multidimensional analysis. The Data Manager then presents this data for processing and manipulation by various Blox. Regardless of the data source, users perform analysis with the same analytical application front end.

The Data Manager architecture enables other databases to be connected through plug-in adapters. Each adapter encapsulates database-specific information for connection and processing. Data source definitions that identify the adapter are stored and administered centrally on DB2 Alphablox. If information for a data source changes, the application administrator changes the information in a single, central location.

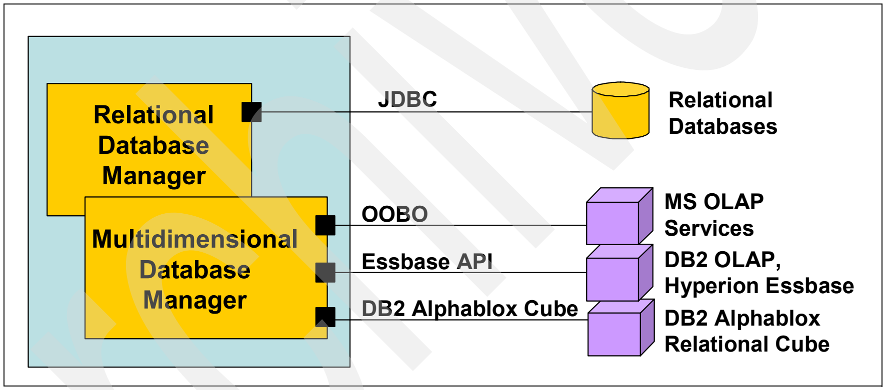


Figure 6-14 DB2 Alphablox data sources

The Data Manager and its associated data adapters provide support for:

- Browsing a collection of predefined data source connections, such as DB2 Alphablox named data sources
- Exposing the available databases within each data source
- Managing the database connections for user sessions
- Translating query objects into the underlying native query language
- Executing queries against the databases
- Modifying result set displays through user-invoked pivoting and drilling
- Using write-back to the underlying data

In addition, the Data Manager allows for traversal of a result set and metadata. Users can retrieve data from the data source, traverse it using the DB2 Alphablox server-side result set and metadata APIs, and then take appropriate action. For instance, the application can be built to use the server-side APIs to traverse through the data looking for a certain condition based on an alert (for example, if actual inventory drops below plan). If the data meets the condition, a process can be started that notifies the affected user (in this case, the buyer for the product). The user can then write-back to the data source (order more product), closing the loop on the specific business process.

Content Manager

The Content Manager handles the setup of applications and examples that exist in the DB2 Alphablox Application Studio library of templates and tools. It has the ability to set up data sources and register applications.

Blox server and client structure

Each Blox has a server-side peer that contains the majority of the Blox functionality. Blox have the ability to render information to the client in a variety of formats. Server-side peers connect to a data source, obtain a result set, and deliver it to the client in the requested runtime format. Client-side components and their server-side peers work together to provide data access, presentation, and manipulation through the built-in user interface of the grid Blox and other presentation Blox.

Using server-side peers, as Figure 6-15 depicts, and client-side components optimizes the performance of DB2 Alphablox analytic-enabled solutions. DB2 Alphablox manages the application logic, separating it from the user interface presentation, thus reducing the burden on the client machine.

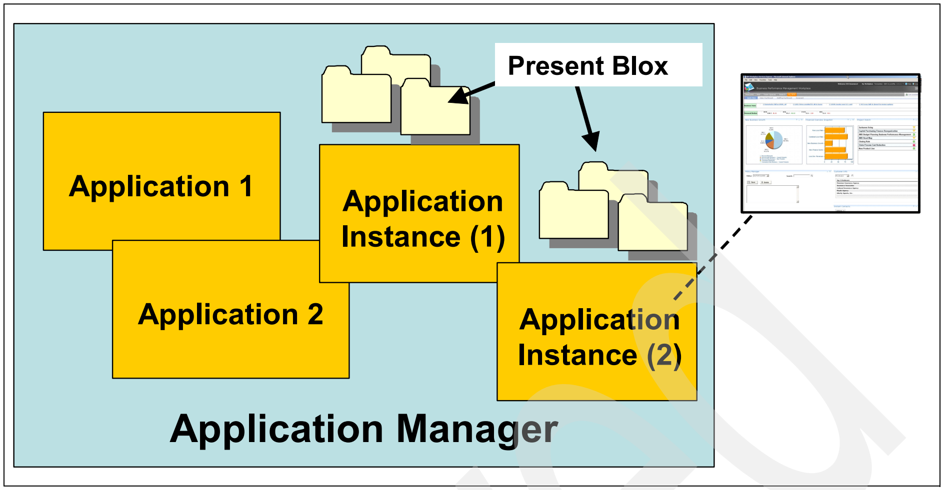


Figure 6-15   DB2 Alphablox server-side peers

Application delivery session flow

Numerous tasks are accomplished between the tiers as an application is accessed, dynamically assembled, and delivered to the client's Web browser. The page processes vary based on the page type and content.

The application server is responsible for the following tasks:

- Network management
- Management of connections
- User authentication and security
- Processing and serving HTML files
- Processing and compiling JSP files using its servlet/JSP engine
- Serving the entire processed page back to the Web browser

DB2 Alphablox is responsible for the following tasks:

- Data access and manipulation
- Dynamically building and deploying the user interface that provides interactive analytic applications
- Managing the data session
- Personalizing the data view

Security

DB2 Alphablox leverages robust security models through the use of J2EE standard APIs (JAAS and JNDI). For example, if the application server uses LDAP, NTLM, Kerberos, or another security mechanism, DB2 Alphablox leverages that mechanism. In addition, DB2 Alphablox leverages the roles that are given to application server users.

DB2 Alphablox users are managed via the application server administration, enabling them to be managed in the same way as users of other applications. This capability allows a developer to construct an application that uses the personalization capability from the application server, combined with information from DB2 Alphablox to dictate the content seen by a user.

By leveraging the application server security mechanisms, DB2 Alphablox developers can implement the security model that works best with their application server. In addition, DB2 Alphablox does not impose additional security constraints. For example, a DB2 Alphablox user ID can be generated by the username passed from the application server.

The only information passed to DB2 Alphablox from the application server is the username. The usernames and passwords are also kept in the DB2 Alphablox repository to help enable personalization and to provide for database access. DB2 Alphablox supports and honors database security. Application users must provide valid database authorization to access application data. When the DB2 Alphablox password is synchronized with the database password, the user also can access the database without a separate sign-on to the database. DB2 Alphablox also works with Secure Sockets Layer (SSL) capabilities, if they exist on any of the servers. SSL encrypts and authenticates all messages sent between the browser, the Web, and the application server pipe.

6.1.2 DWE OLAP

DWE OLAP is a feature of DWE. Combined with leading BI infrastructure solutions, DWE OLAP accelerates the development and lowers the total cost of ownership of OLAP applications. It allows you to create specialized relational structures within DB2 UDB that expose metadata to any BI software in the market. This ready to use metadata integration makes it easier for business analysts to deploy and manage BI solutions, shortening time to market, reducing development costs, and streamlining IT administration.

With DWE OLAP, no extensive knowledge of OLAP is required to accelerate real-time data analysis. Database administrators (DBAs) can drag multiple objects onto predefined layouts to quickly add OLAP function to the data warehouse. And they can use DWE OLAP to create summary tables and metadata to enable faster data access by Business Partner tools. DBAs can also develop associated SQL queries so users can start with the summarizations of the cube and then drill down into more customized detail.

DWE OLAP:

- Accelerates DB2 UDB queries by recommending MQTs
- Enables the DB2 UDB optimizer to rewrite incoming queries to access MQTs
- Loads cubes, performs drill-through queries and ad hoc analysis directly, using the relational tables in DB2 UDB

DWE OLAP includes features and functions that transform the IBM DB2 Universal Database system into a platform for managing and deploying multidimensional data across the enterprise. You can create a set of metadata objects to dimensionally model your relational data and OLAP structures naturally, productively, and efficiently. DWE OLAP stores each of the metadata objects that you create in the DB2 UDB catalog.

In addition, DWE OLAP includes the OLAP Center with which you can create, manipulate, import, or export cube models, cubes, and other metadata objects for use in OLAP tools. The OLAP Center provides easy-to-use wizards and windows to help you work with your metadata. For example, the Optimization Advisor analyzes the metadata and recommends how to build summary tables that store and index aggregated data for your OLAP-style SQL queries.

You can use DWE OLAP to streamline the deployment and management of OLAP solutions and improve the performance of OLAP tools and applications.

With DWE OLAP, you can describe the dimensional structure of your relational tables and create OLAP constructs. You can also store the structural information and the OLAP constructs as multidimensional metadata in the DB2 database.

The new multidimensional metadata in DB2 UDB provides two major benefits:

- Improves the flow of the multidimensional metadata between business intelligence tools and applications. For example, you can use a graphical interface that is provided to store the multidimensional metadata as part of the DB2 database and make that metadata available for all warehousing tools and business intelligence applications.
- Enhances the performance of OLAP-style queries. Based on the multidimensional metadata, you can create DB2 summary tables using the recommendations from the Optimization Advisor. The summary tables contain precalculated data that maps to your OLAP structures. Queries that are generated from the warehousing or business intelligence application with the same OLAP structure gain performance improvement.

DWE OLAP exploits DB2 features such as summary tables, different index schemes, OLAP-style operators, and aggregation functions. The following components are provided:

- Multidimensional metadata objects: You can create a set of metadata objects to dimensionally model your relational data and OLAP structures. DWE OLAP stores each of the metadata objects that you create in the DB2 catalog.
- OLAP Center: With the OLAP Center, you can create, manipulate, import, or export cube models, cubes, and other metadata objects to be used in OLAP tools. The OLAP Center provides easy-to-use wizards and windows to help you work with your metadata objects. For example, the Optimization Advisor analyzes your metadata objects and recommends how to build summary tables that store and index aggregated data for your OLAP-style SQL queries.
- Multidimensional Services: DWE OLAP provides an SQL-based and XML-based application programming interface (API) for OLAP tools and application developers. Using CLI, ODBC, or JDBC connections, or by using embedded SQL to DB2 UDB, applications and tools can use a single stored procedure to create, modify, and retrieve metadata objects.
- Sample data: A sample application and database are available to help you learn how to use the product.

You can also exchange metadata objects between the DB2 catalog and OLAP tools. To import or export metadata objects to or from the DB2 catalog, utilities called *metadata bridges* are available for specific OLAP and database tools. See the documentation for your particular OLAP or database tool to determine if a metadata bridge is provided.

DWE OLAP metadata objects describe relational tables as OLAP structures, but these metadata objects are different from traditional OLAP objects. Metadata objects store metadata about the data in the base tables, they describe where pertinent data is located, and they describe relationships within the base data.

DWE OLAP stores information about your relational data in metadata objects that provide a new perspective from which to understand your data. DWE OLAP extends the DB2 catalog so that in addition to storing information about tables and columns, the DB2 catalog contains information about how the tables and columns relate to OLAP objects and the relationships between those metadata objects.

Some metadata objects act as a base to directly access relational data by aggregating data or directly corresponding to particular columns in relational tables. Other metadata objects describe relationships between the base metadata objects and link the base metadata objects together. All of the metadata objects can be grouped by their relationships to each other into a metadata object called a *cube model*. Essentially, a cube model represents a particular grouping and configuration of relational tables.

6.1.3 DWE Design Studio

DWE Design Studio is an IDE for data warehouse projects, which integrates consistent and interoperable tools for:

- Connecting and browsing databases
- Exploring data
- Designing physical database models (reverse/forward engineering)
- Designing OLAP objects
- Designing and deploying Data Mining flows
- Designing and deploying SQL Warehousing data and control flows

As an Eclipse-based tool, DWE Design studio is easily extended with third party tools.

Eclipse is a universal platform for integrating tools, providing the powerful framework and the common GUI and infrastructure required to integrate tools. You can easily extend this platform by installing plug-ins, which extend defined extension points provided by the platform itself or by other plug-ins.

The framework and general workbench of Eclipse are developed and maintained by a large community of companies, including IBM.

Benefits of using Eclipse, in addition to those already mentioned, are:

- User experience is similar to all other Eclipse-based products, such as WebSphere Studio Application Developer and Rational Application Developer.
- As an open-source project, there are resources on the Internet explaining how to extend the platform and write tools for it.
- Third-party tools are already in the market and available to use, delivering an extensive portfolio.

DWE Design Studio aggregates tools to handle SQL tasks, Mining Editors, OLAP Tools, Data Exploration Tools, and more as Figure 6-16 depicts.

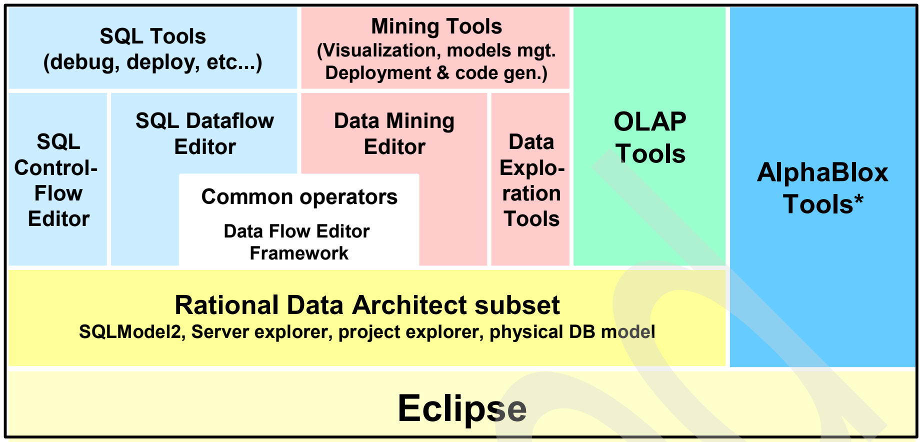


Figure 6-16 DWE Design Studio overview

The DWE Design Studio is a workbench composed of perspectives. The look and feel of this workbench is pretty much the same as the other tools based on Eclipse, and Figure 6-17 on page 151 depicts this look and feel.

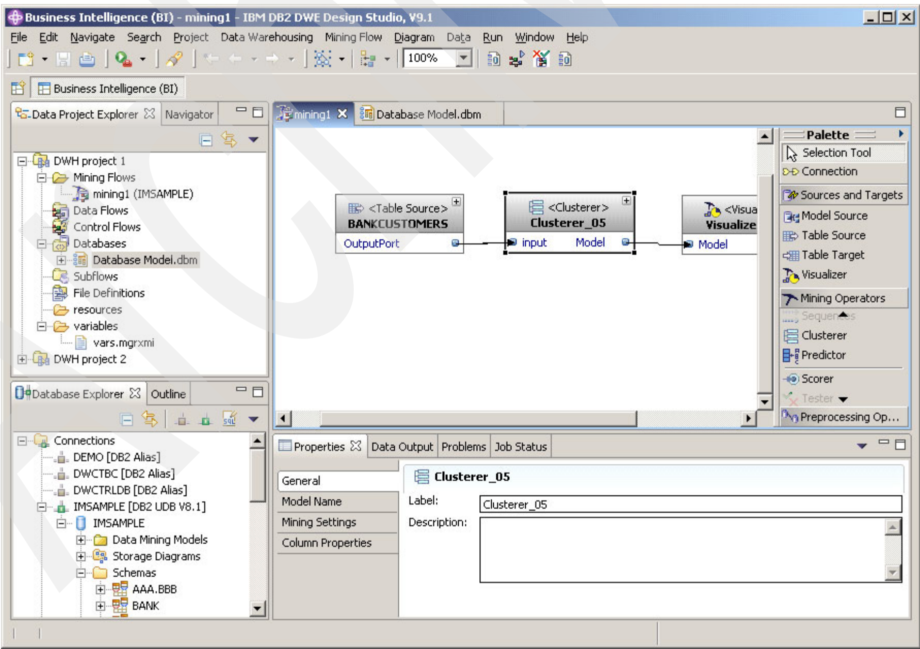


Figure 6-17 DWE Design Studio workbench

DWE Design Studio perspectives are arrangements of views and editors with different perspectives that are suited for different tasks and which deliver the right tool for a specific job.

Although the deployed standard perspectives are very comprehensive, they might not have all the features that you need. To solve that, perspectives are live entities which can be customized as needed. As mentioned before, perspectives are aggregation of functions that are normally used to perform one activity. As a live entity, if you need a feature that is not shown or enabled on the perspective, you can customize the deployed perspective.

Business Intelligence perspective

In this perspective, all DWE-related activities are aggregated, as you can see in Figure 6-18 on page 152.

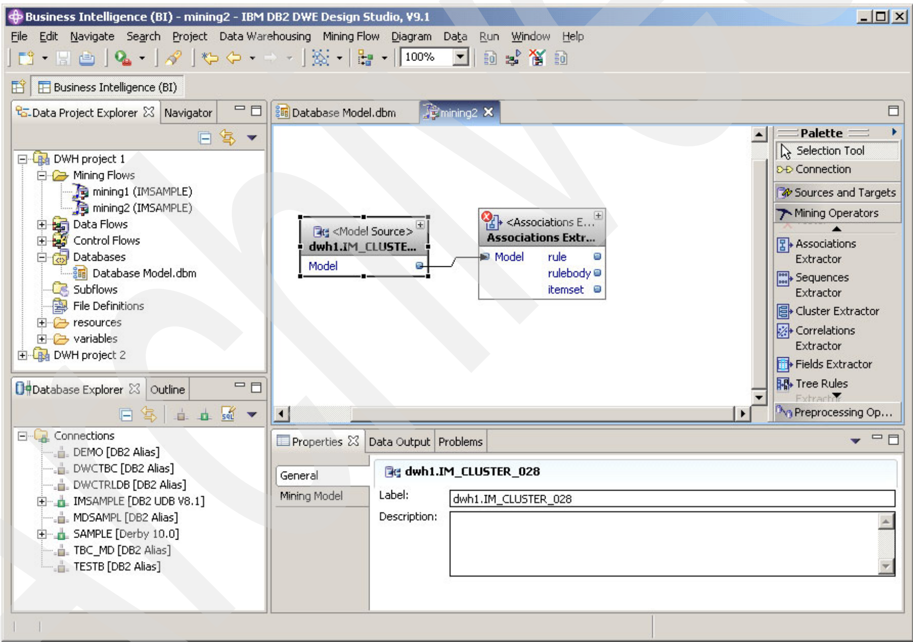


Figure 6-18 Business Intelligence perspective

In the business intelligence perspective, the outline view shows an overview about the structure of the document which is currently edited. The appearance of the view depends on the nature of the edited document.

When the flow is edited, the outline offers two representations:

- A graphical representation showing the entire flow.

- A tree representation where the objects composing the flow can be easily selected.

Figure 6-23 on page 155 depicts these representations.

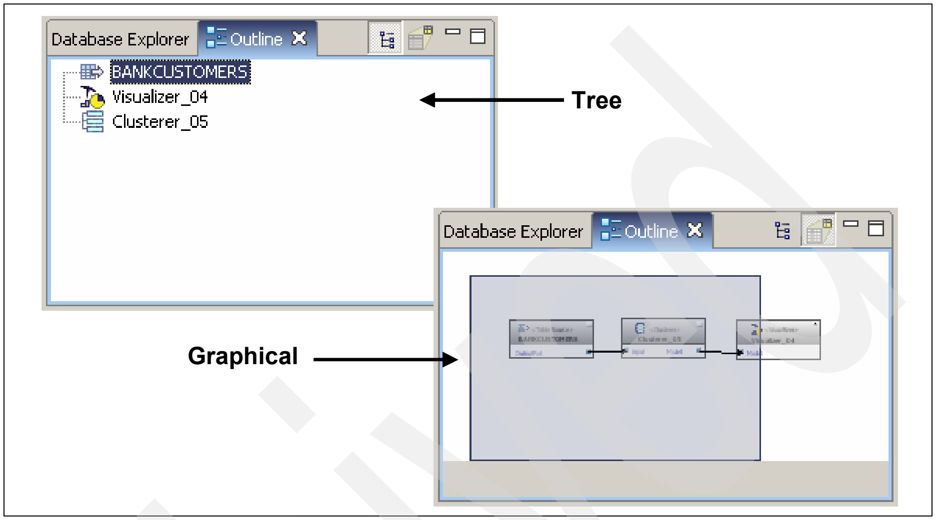


Figure 6-19 Outline view representations: Tree and Graphical

The properties view allows you to view and modify the properties of the current selected object. The appearance of this view is dependent on the current selection.

This view is one of the most important views inside the BI perspective when designing flows or database objects, and you can see this view in Figure 6-20.

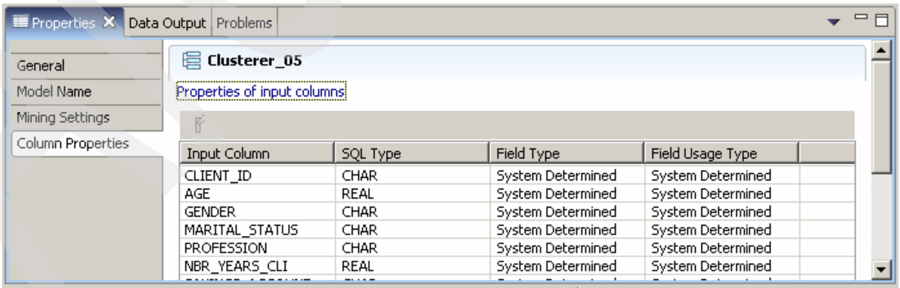


Figure 6-20 Properties view

The Data Output view displays the result of SQL statements when they are executed on a database. This view is used in the following scenarios:

- Inspect the content of a table
- Execute data mining flows
- Execute SQL/DDL scripts or to perform any operation on the database

The history of the last run queries is kept on the left-hand side of the view.

The right-hand side of the view displays a sampling of an eventual result set returned by the query and, under messages, the full SQL statement that has been executed, along with the eventual messages returned by the database. Figure 6-21 depicts this view.

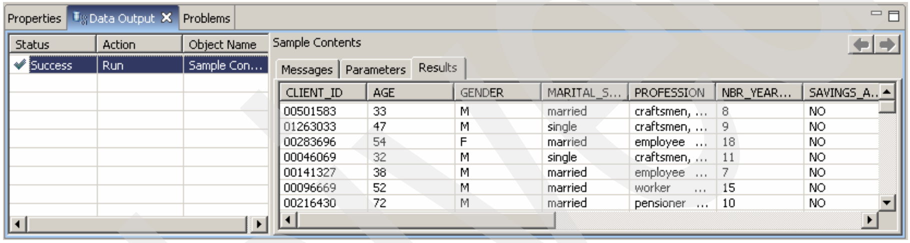


Figure 6-21 Data output view

Still in the business intelligence perspective, there is a problems view. This view shows eventual errors and warnings detected during validation of the resources contained in the opened projects. There are three levels of severity messages: Errors, Warnings, and Information.

You can sort the messages by severity, description, or resources. To check the message, click it and the resource containing the problem opens. An example of this view is in Figure 6-22.

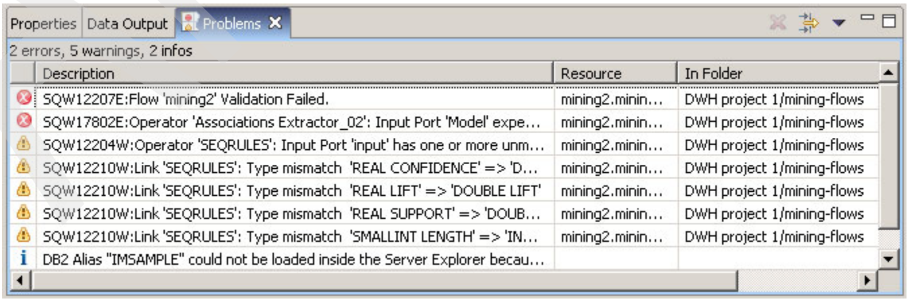


Figure 6-22 Problems view

Data Modeling

Using the enabled features, you can create new database designs or use reverse engineering. You can create diagrams from existing schemata, such as UML notations, and modify entities graphically.

Another enabled feature is the ability to compare database objects with objects created on an existing database. Figure 6-23 depicts a sample project using this feature.

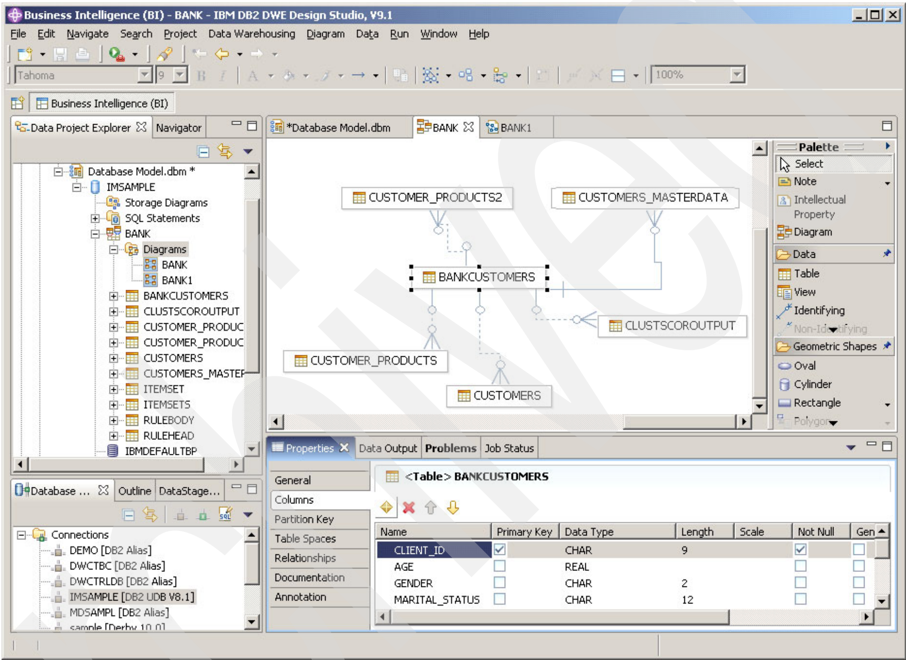


Figure 6-23 Data Modeling perspective

In this perspective, you can also manage physical entities, such as tablespaces and containers. Figure 6-24 on page 156 depicts this perspective.

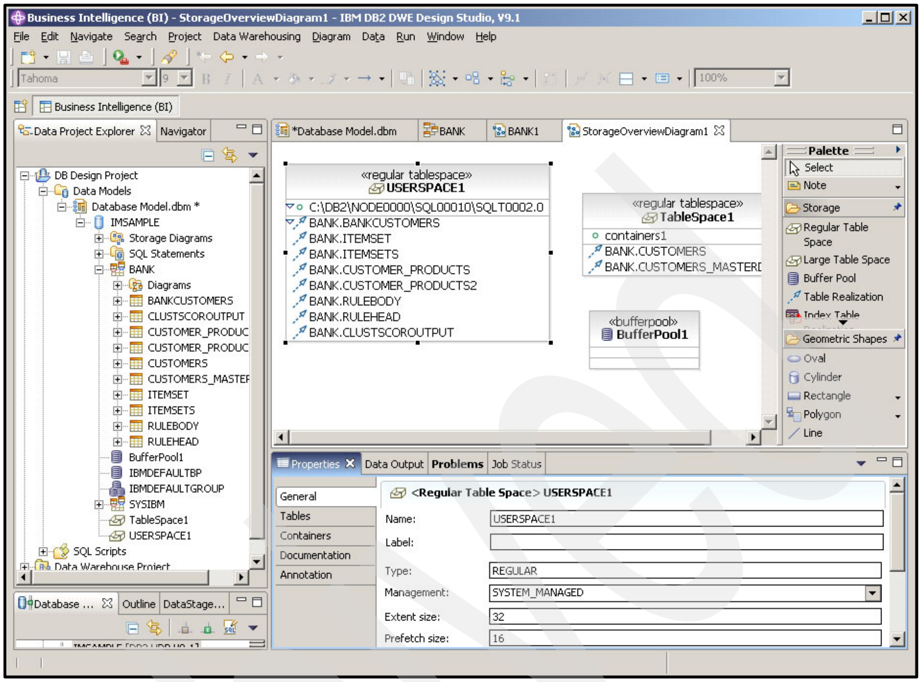


Figure 6-24 Physical modeling

OLAP modeling is an extension of the data modeling capabilities used to design OLAP cubes in Physical Data Models. Figure 6-25 on page 157 depicts a sample.

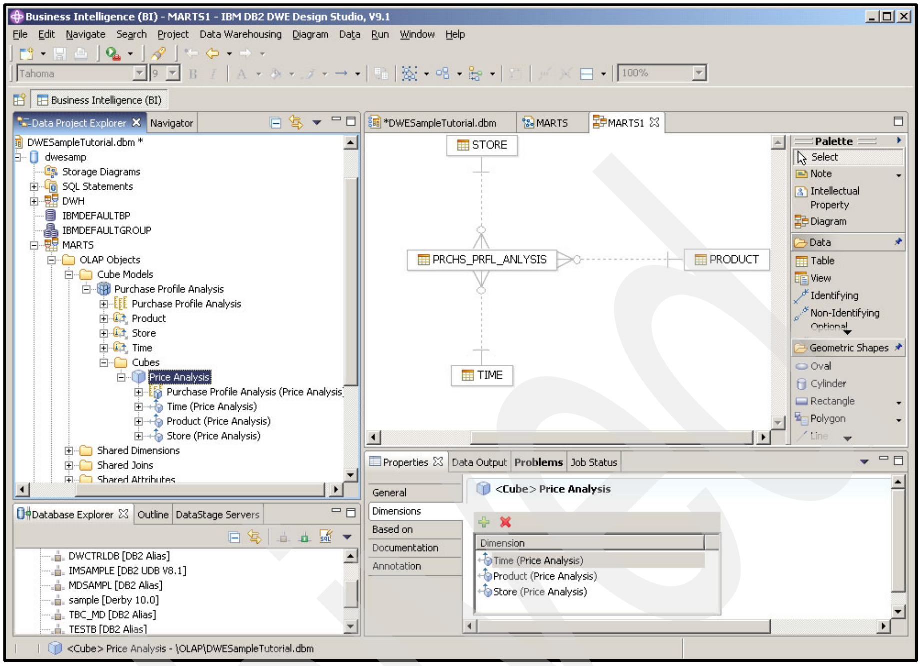


Figure 6-25 OLAP modeling

DWE SQL Warehousing

SQL Warehousing is used to control and generate deployable packages that are composed of data flows and control flows. A *data flow* is used to define data transformation steps through a library of operations used for common data extraction and transformation steps. For more comprehensive operations, this library can be extended.

You can use general SQL operators to directly express transformations in SQL and also create reusable subflows for often used transformation patterns, helping to decrease complexity and maintenance.

Figure 6-26 on page 158 depicts a sample data flow.

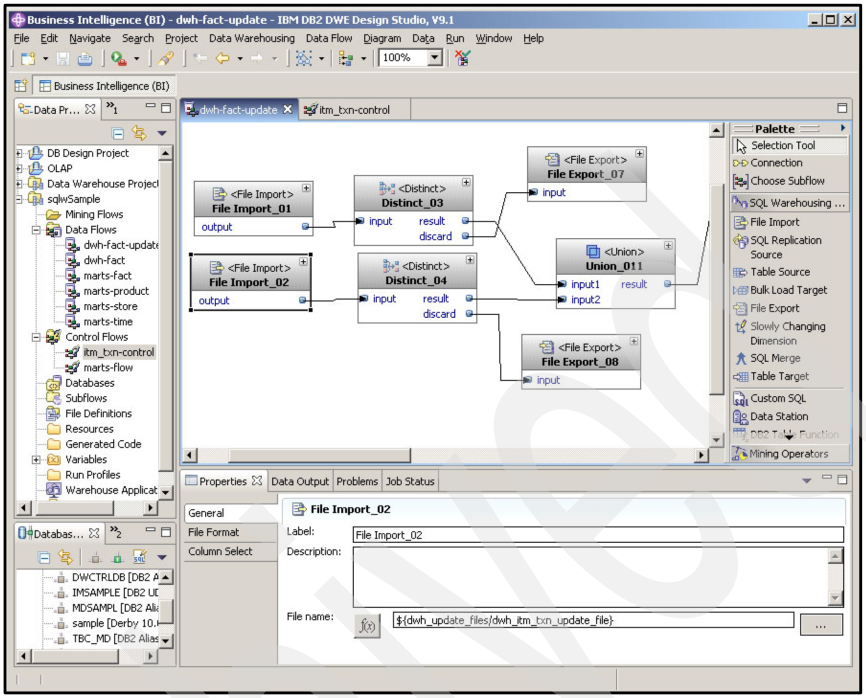


Figure 6-26 Sample data flow

Another type of flow, also generated by DWE Design Studio, is used to control and coordinate the execution of several data flows.

In this flow, you have support for execution conditions, such as *on success*, *on failure*, and *always,* to create logical flows.

Figure 6-27 on page 159 depicts a sample control flow.

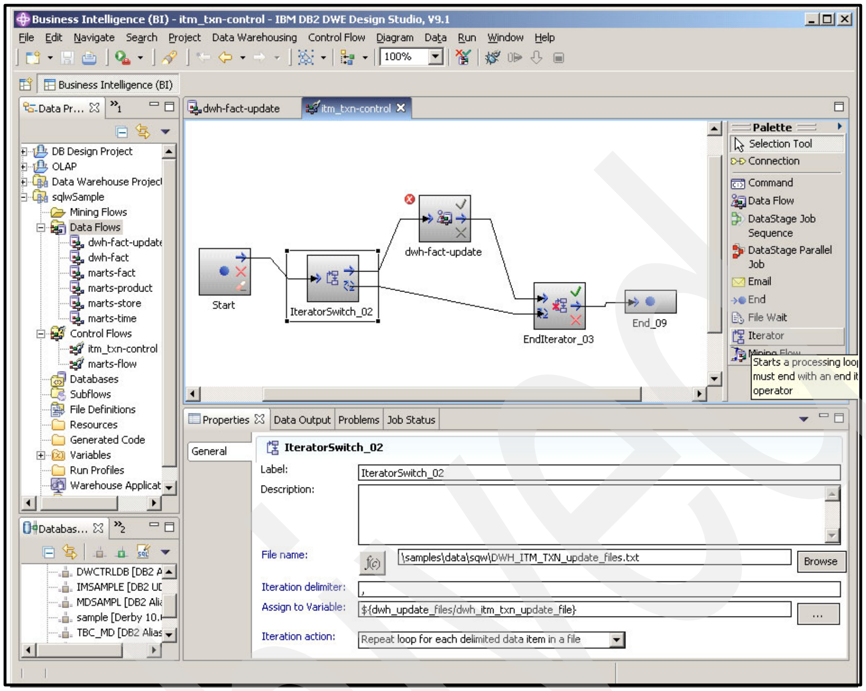


Figure 6-27 Sample control flow

For more information about data warehousing features, see 6.1.9, “DWE SQL Warehousing Tool” on page 173.

Projects

Before you can design data warehouse resources, you have to create a project. The most important types of projects for data warehouse development are:

- Data Design: Design database physical models, including OLAP modeling, to execute forward/reverse engineering over database models.
- Data Warehouse: Design SQL Warehouse and data mining flows. This project type is also used to generate Data Warehouse Application packages to deploy on DWE servers.

You can see a project creation sample in Figure 6-28 on page 160.

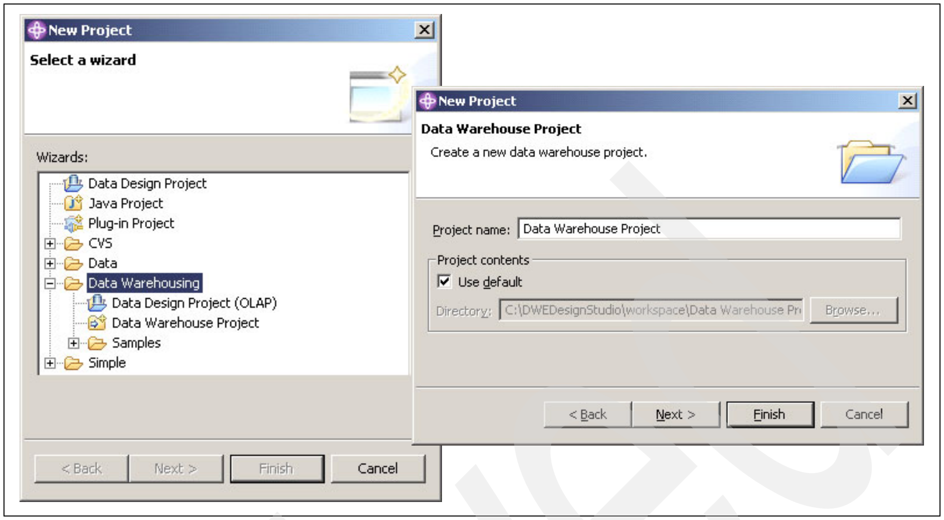


Figure 6-28 Creating a project

6.1.4 DWE Integrated Installer

DWE Integrated Installer is responsible for installing numerous products from one interface.

Figure 6-29 on page 161 shows the first window of DWE Integrated Installer. Note that you can select between Enterprise Edition and Base Edition.

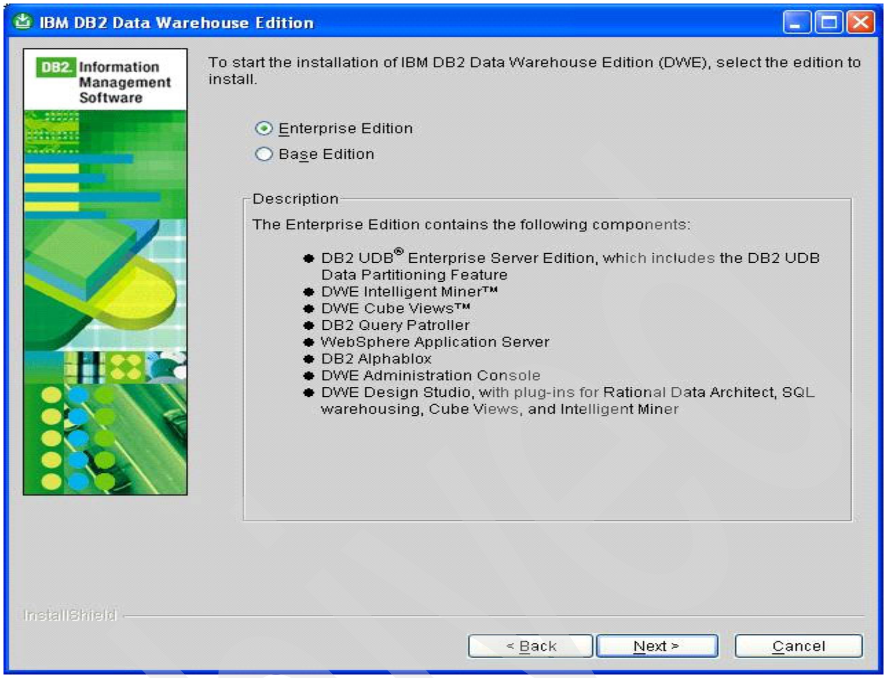


Figure 6-29 Integrated installer

Through the use of this tool, you can select several features that you can use when you have distinct servers for specific roles on the business intelligence scenario. You can also customize the components that you use on the working server. Figure 6-30 on page 162 depicts the available features.

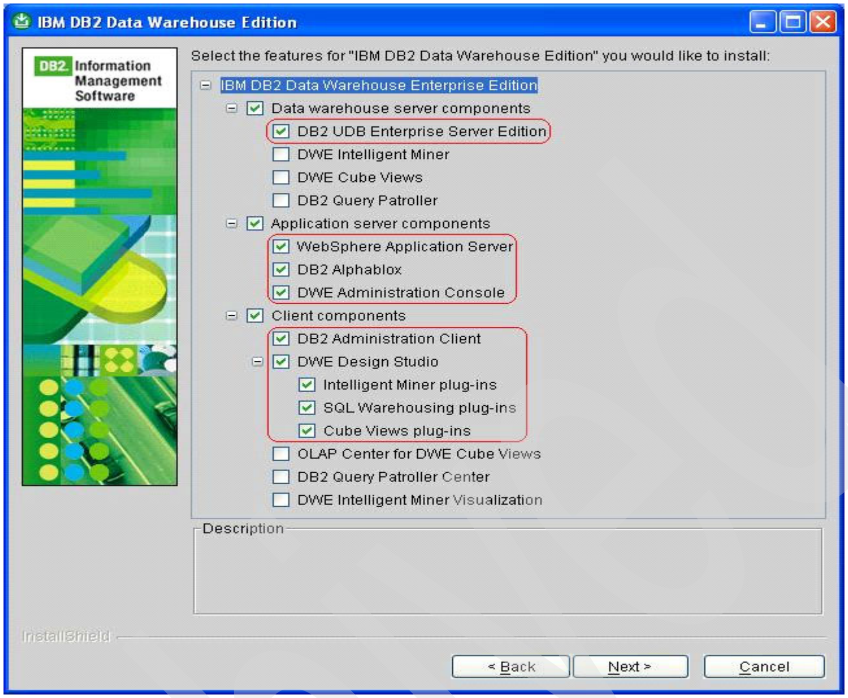


Figure 6-30 Customizing installation features

6.1.5 DB2 data partitioning feature

A *database partition* is part of a database that consists of its own data, indexes, configuration files, and transaction logs. It is sometimes called a *node* or a *database node*.

A *partitioned database* is a database with two or more partitions. Tables can then be located in one or more database partitions. Processors associated with each database partition are used to satisfy table requests. Data retrieval and update requests are decomposed into sub-requests and executed in parallel among the applicable database partitions. In this type of database, data is hashed for storage.

Tables can be located in one or more database partitions. When table data is distributed across multiple partitions, some of its rows are stored in one partition, and other rows are stored in other partitions. Data retrieval and update requests are decomposed into sub-requests and executed in parallel among the applicable database partitions. The fact that databases are split across database partitions is transparent to users.

A single database partition might exist on each physical component that makes up the computer. The processors on each system are used by the database manager at each database partition to manage its part of the total data in the database. Because data is divided across database partitions, you can use the power of multiple processors on multiple computers to satisfy requests for information.

Data retrieval and update requests are decomposed into sub-requests and are executed in parallel among the applicable database partitions.

User interaction occurs through one database partition, known as the coordinator partition for that user. The coordinator runs on the same database partition as the application, or, in the case of a remote application, the database partition to which that application is connected. Any database partition can be used as a coordinator partition.

DPF allows great flexibility in spreading data across multiple partitions (nodes) of a partitioned database. Users can choose how to partition their data by declaring partitioning keys and can determine which and how many partitions their table data can be spread across by selecting the database partition group and table space in which the data should be stored. In addition, a partitioning map (which is updateable) specifies the mapping of partitioning key values to partitions. This makes it possible for flexible workload parallelization across a partitioned database for large tables, while allowing smaller tables to be stored on one or a small number of partitions if the application designer so chooses. Each local partition can have local indexes on the data it stores to provide high performance local data access.

A partitioned database supports a partitioned storage model, in which the partitioning key is used to partition table data across a set of database partitions. Index data is also partitioned with its corresponding tables and stored locally with each partition.

Before partitions can be used to store data, they must be defined to the database manager. Partitions are defined in a file called db2nodes.cfg.

The partitioning key for a table in a table space on a partitioned database partition group is specified in the CREATE TABLE statement or the ALTER TABLE statement. If not specified, a partitioning key for a table is created by default from the first column of the primary key. If no primary key is defined, the default partitioning key is the first column defined in that table that has a data type other than a long or a LOB data type. Partitioned tables must have at least one column that is neither a long nor a LOB data type. A table in a table space that is in a single partition database partition group has a partitioning key only if it is explicitly specified.

Hash partitioning is used to place a row in a partition as follows:

- A hashing algorithm (partitioning function) is applied to all of the columns of the partitioning key, which results in the generation of a partitioning map index value.
- The partition number at that index value in the partitioning map identifies the partition in which the row is to be stored.

DB2 DPF supports *partial declustering*, which means that a table can be partitioned across a subset of partitions in the system (that is, a database partition group). Tables do not have to be partitioned across all of the partitions in the system.

DB2 DPF has the capability to recognize when data being accessed for a join or a subquery is located at the same partition in the same database partition group. This is known as *table collocation*. Rows in collocated tables with the same partitioning key values are located on the same partition. DB2 DPF can choose to perform join or subquery processing at the partition in which the data is stored. This can have significant performance advantages.

Collocated tables must:

- Be in the same database partition group, one that is not being redistributed. (During redistribution, tables in the database partition group can use different partitioning maps - they are not collocated.)
- Have partitioning keys with the same number of columns.
- Have the corresponding columns of the partitioning key be partition compatible.
- Be in a single partition database partition group defined on the same partition.

6.1.6 DB2 Query Patroller

DB2 Query Patroller is a powerful query management system that you can use to proactively and dynamically control the flow of queries against your DB2 database in the following key ways:

- Define separate query classes for queries of different sizes to better share system resources among queries and to prevent smaller queries from getting stuck behind larger ones.
- Give queries submitted by certain users high priority so that these queries run sooner.
- Programmatically put large queries on hold so that they can be canceled or scheduled to run during off-peak hours.
- Track and cancel runaway queries.

The features of Query Patroller allow you to regulate the database query workload so that small queries and high-priority queries can run promptly and your system resources are used efficiently. In addition, information about completed queries can be collected and analyzed to determine trends across queries, heavy users, and frequently used tables and indexes.

Administrators can use Query Patroller to:

- Set resource usage policies at the system level and at the user level.
- Actively monitor and manage system usage by canceling or rescheduling queries that could impact database performance.
- Generate reports that assist in identifying trends in database usage, such as which objects are being accessed and which individuals or groups of users are the biggest contributors to the workload.

Query submitters can use Query Patroller to:

- Monitor the queries they have submitted.
- Store query results for future retrieval and reuse, effectively eliminating the need for repetitive query submission.
- Set a variety of preferences to customize their query submissions, such as whether to receive e-mail notification when a query completes.

Query Patroller components

DB2 Query Patroller is a client/server solution consisting of the following components:

- Query Patroller server
- Query Patroller Center
- Query Patroller command line support

DB2 Query Patroller can be deployed on a system running DB2 Enterprise Server Edition.

Query Patroller server

When you install Query Patroller server, the following software elements are deployed to the target computer:

- **Query Patroller stored procedures:** Query Patroller stored procedures are called by other Query Patroller components to perform the necessary database tasks.
- **Control tables**: When Query Patroller is set up to manage queries issued against a database, the DB2QP schema, control tables, triggers, functions, and procedures are created within that database. The control tables store all

of the information that Query Patroller requires to manage queries. This information includes the following:

 – Query Patroller system properties settings
 – Query class information
 – Submitter information, including query submission preferences
 – Operator information
 – Managed query properties information
 – Historical query properties information
 – Query result information
 – Historical analysis data
 – Scheduled purge job details

 For example, the SUBMITTER_PROFILE table contains information such as the submitter's ID, authority level, and the maximum number of queries that the user can have running simultaneously. When the user submits a query, Query Patroller references the SUBMITTER_PROFILE table for these parameters.

- **Log files**: Diagnostic information about errors is recorded in four different Query Patroller log files:
 – qpsetup.log
 – qpmigrate.log
 – qpuser.log
 – qpdiag.log

Query Patroller Center

The Query Patroller Center is a graphical user interface that allows administrators to manage Query Patroller system properties, users, and queries and to view historical analysis reports. The Query Patroller Center also allows query submitters to manage their queries, save query results, and customize their query submission preferences.

The look and functionality of the Query Patroller Center varies depending on different factors, such as the authority of the user and whether the DB2 administration tools are also installed.

An administrator has access to the full functionality of Query Patroller Center. The following list shows some of the tasks that administrators can do with Query Patroller Center:

- Manage the Query Patroller system parameters.
- Create, update, or delete profiles for Query Patroller submitters and operators.
- Create, update, or delete submission preferences for Query Patroller submitters.
- Create, update, or delete query classes.
- Monitor and manage queries that have been intercepted by the Query Patroller system.
- Generate and analyze reports that display database usage history.

A submitter has access to a subset of the functionality of Query Patroller Center. The following list shows some of the tasks that submitters can do with Query Patroller Center:

- Monitor and manage queries that they have submitted through the Query Patroller system.
- Store results of the queries that they have submitted for future retrieval.
- Show or file results of the queries that they have submitted.
- Create, update, or delete their own query submission preferences.

Query Patroller command line support

Command line support enables Query Patroller administrators and submitters to perform most Query Patroller tasks from the DB2 CLP or from the operating system's command line prompt. Query Patroller commands can also be combined with shell scripts or languages such as Perl, awk, and REXX.

6.1.7 DWE Mining

Data Mining is embedded into DB2 through the use of DB2 stored procedures and user-defined functions (UDFs). That are three distinct modules:

- Modeling
- Scoring
- Visualization

A typical data modeling process is based on the steps depicted in Table 6-2 on page 168. As a prerequisite, data should have been preprocessed.

Table 6-2 Common mining steps

Step	Module	Description
1	Modeling	Defining a mining task
2	Modeling	Doing a mining run and building a model
3	Visualization	Visualizing the model
4	Scoring	Scoring new data against the model (prediction)

The data mining run is accomplished by:

- DWE Mining Editor
- Easy Mining Procedures
- SQL Stored Procedures

Through the use of the DWE Mining Editor, you can compose mining tasks as an integrated part of end-to-end data flows using a graphical canvas.

The Easy Mining interface is a high level API composed of Java UDFs with a simplified interface to do a mining run with a single call to an SQL stored procedure.

SQL Stored Procedures with SQL/MM API is a detailed expert API composed of UDFs for the mining tasks, models, and test results. UDMs are used for defining data mining tasks and stored procedures to build and test models. UDFs, and table UDFs, are used for analyzing built models such as model signature, model statistics, and model quality.

An example of the use of these features is depicted in Figure 6-31 on page 169.

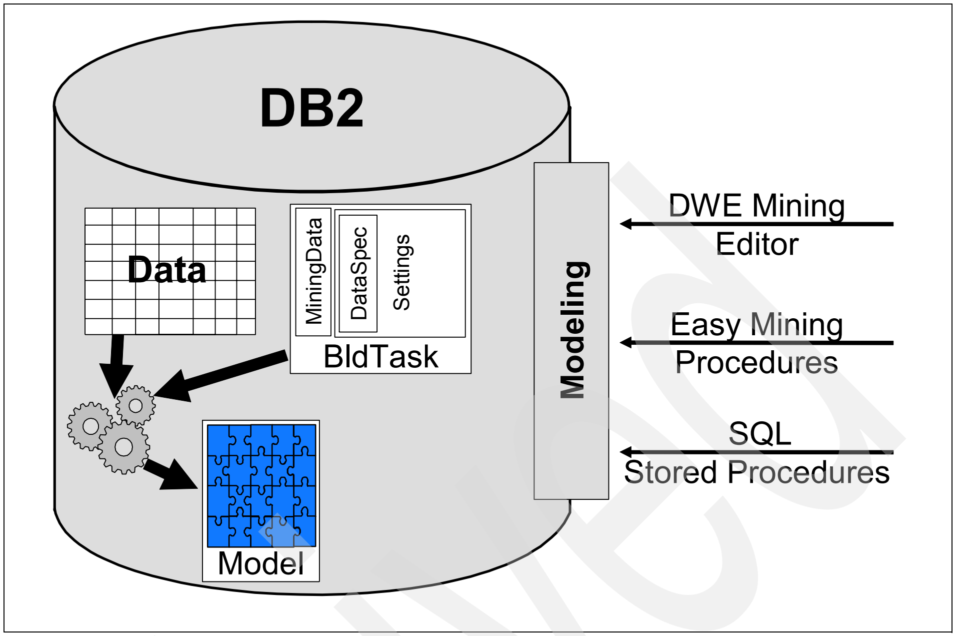


Figure 6-31 Data mining run

The visualization functionality uses Java visualizer for Predictive Model Markup Language (PMML) for full functionality with models created by DWE V9.1 and IBM DWE Mining. PMML models from other providers can be visualized. Currently, the PMML model versions supported are V2.1, V3.0, and V3.1.

You can read and modify models stored as files in PMML format and in DB2 as objects of the SQL/MM types. Visualization is available as a stand-alone application as part of DWE V9.1 or a Java Applet on an HTTP Server. A Visualization sample can be seen in Figure 6-32 on page 170.

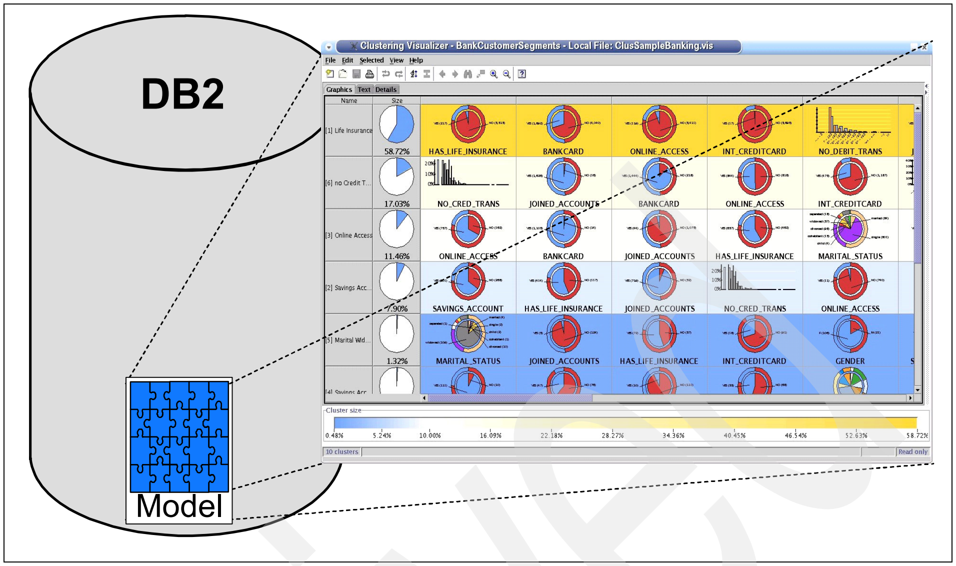


Figure 6-32 Visualizing a model

The Scoring functionality is accomplished by the use of the Easy Mining Interface, which is based on Java UDFs to do a scoring run with a single call to an SQL stored procedure and store the scoring result in a view.

Scoring can also be accomplished with SQL/MM APIs by the use of UDTs for the models, result specs, and scoring results and UDFs to import models into DB2, score data against models, and analyze the scoring result.

An example of Scoring is depicted in Figure 6-33 on page 171.

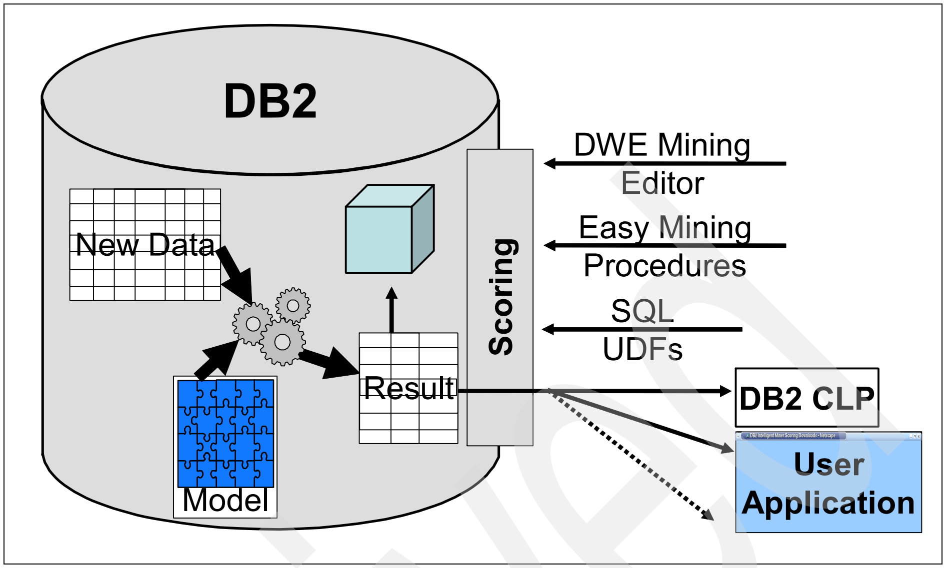


Figure 6-33 Scoring new data against a model

6.1.8 DWE Administration Console

The DWE Administration Console (Admin Console) is a J2EE application that runs on an application server. It is developed in Java and based on Java Server Faces (JSF).

The DWE Admin Console, is a common interface for administrative tasks which provides a unified *one-stop-shopping* point of entry and enables remote access via a Web-based client. Its positioning in the DWE structure is depicted in Figure 6-34 on page 172.

Some features that can be executed on the DWE Admin Console are:

- Management and monitoring of SQL Warehousing applications
- Management and optimization of OLAP Cube models
- Management and Scoring Data Mining

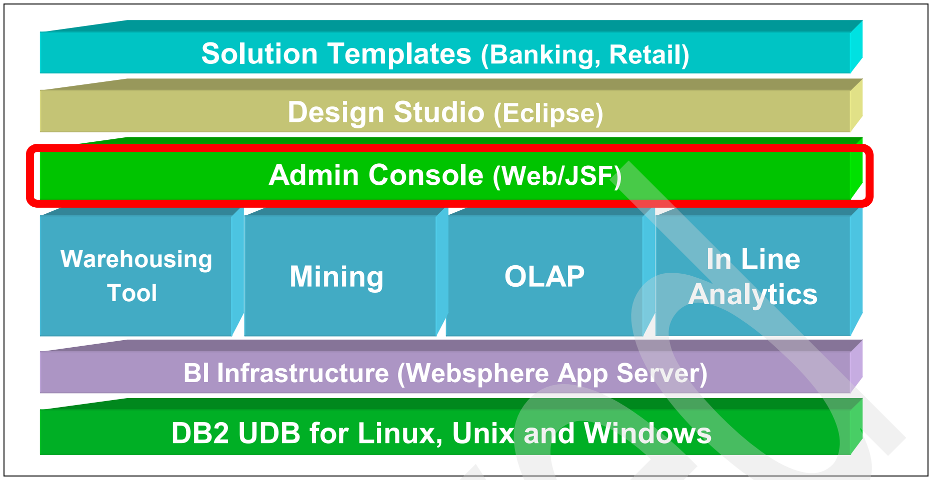


Figure 6-34 DWE Admin Console positioning

As a J2EE application hosted on an application server, you can reach role-based security resource management and manage Enterprise Java Beans. The DWE Admin Console components are depicted in Figure 6-35.

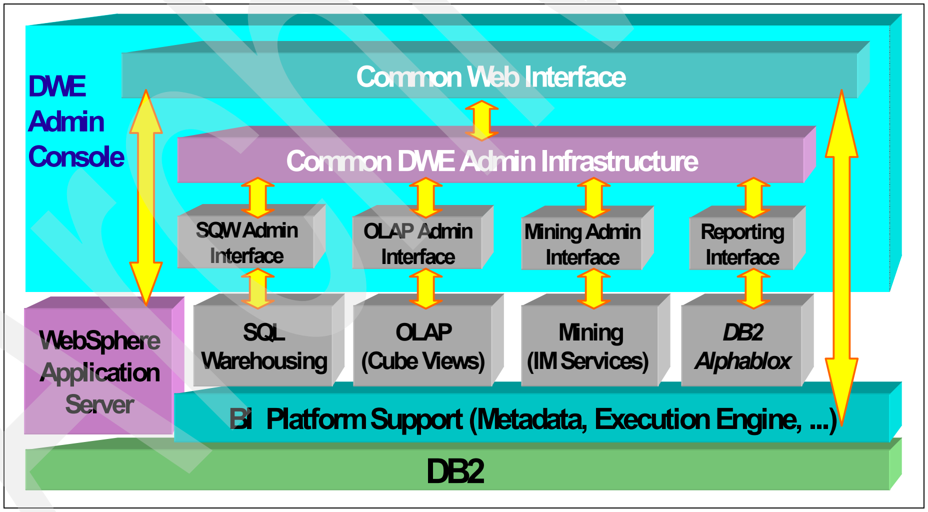


Figure 6-35 DWE Admin Console components

DWE Admin Console resides on the same system as the WebSphere Application Server and is deployed to WebSphere Application Server as an enterprise application, which leverages WebSphere technology to access resources such as databases, scheduler, work manager, and mail provider.

DWE Data Integration Service (DIS) can access the data sources via WebSphere application interface or DB2 application interface.

As depicted in Figure 6-36, the execution, source, and target databases referenced by data flow activities can be local or remote to the DWE Admin Console and connections to these databases can be managed by WebSphere or DIS.

The target databases for SQL script activities can also be local or remote to the DWE Admin Console. The connections to these target databases are managed by DIS, and the control database for DWE can reside locally or remotely.

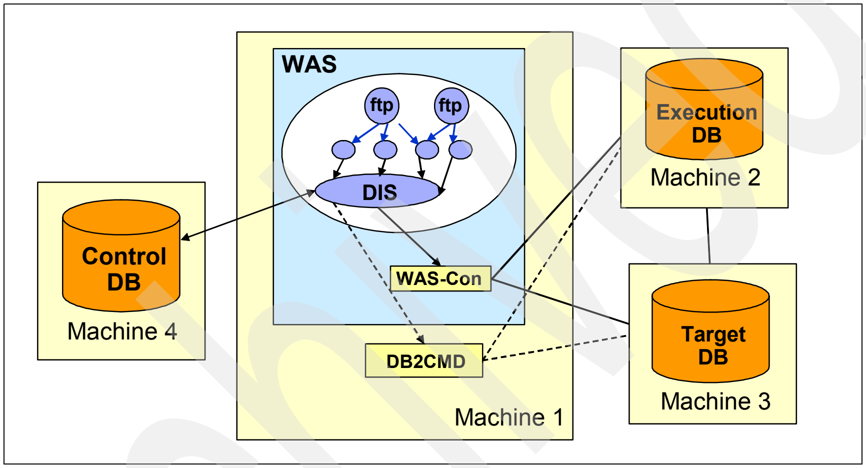


Figure 6-36 DWE Admin Console topology

6.1.9 DWE SQL Warehousing Tool

DWE SQL Warehousing Tool (SQW) can be used to build and maintain data warehouses in DB2 contexts as well as components and configuration. Some features and functions of SQLW are:

- Interaction with physical modeling through the use of DWE Design Studio, OLAP, and mining functions in an integrated Eclipse-based GUI
- Code and execution plan generation optimized for DB2
- In-database (DB2) SQL transformations and updates for data in warehouse and mart tables
- WebSphere Application Server runtime environment for role-based application management (deployment, scheduling, and monitoring) via the DWE Admin Console
- Integration points with WebSphere DataStage ETL engine

Components you need to create SQLW are:

- DWE Design Studio
- DWE Admin Console
- DB2 UDB
- WebSphere Application Server

The life cycle of a data warehouse application using SQLW can be segmented as shown in Table 6-3.

Table 6-3 Life cycle of a data warehouse application steps

Step	Design Studio	Admin Console	DB2 Instance	WAS	Description
1	Needed	Needed	Needed	Needed	Install design and runtime environments.
2	Needed	Not needed	Needed	Not needed	Design and validate data flows.
3	Needed	Not needed	Needed	Not needed	Test run data flows.
4	Needed	Not needed	Needed	Not needed	Design and validate control flows.
5	Needed	Not needed	Needed	Not needed	Test run control flows.
6	Needed	Not needed	Not needed	Not needed	Prepare control flow application for deployment.
7	Not needed	Needed	Needed	Needed	Deploy application.
8	Not needed	Needed	Needed	Needed	Run and manage application at process (control flow) level.
9	Needed	Needed	Needed	Needed	Iterate based on changes in source and target databases.

Data Flows

Data flows are models that translate data movement and transformation requirements into repeatable, SQL-based warehouse building processes. Data from source files and tables moves through a series of transformation steps before loading or updating a target file or table.

The example depicted in Figure 6-37 selects data from a DB2 staging table, removes duplicates, sorts the rows, and inserts the result into another DB2 table. Discarded duplicates go to a flat file.

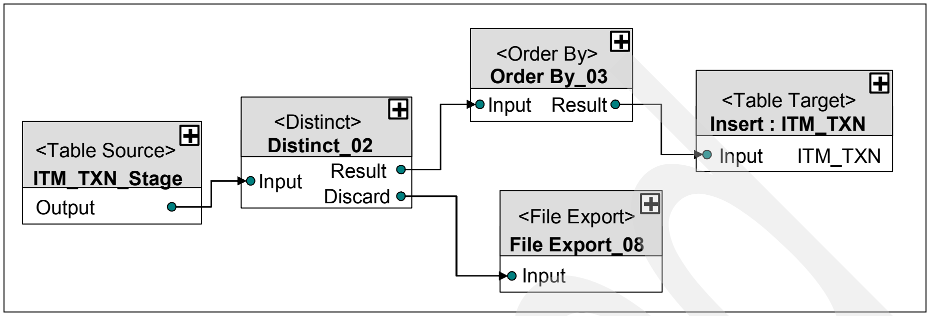


Figure 6-37 simple data transformation example

The possible sources and targets for data flows are:

- Sources
  - File import
  - Table source
  - SQL replication source
- Targets
  - File export
  - Table target (insert, update)
  - Bulk load target (DB2 load utility)
  - SQL merge
  - Slowly changing dimension (SCD)
  - Data station (special staging operator, intermediate target)

To execute data flows in DWE Design Studio, you must:

- Have no errors listed on the problem view.
- Choose or define the run profile.
- Select resources and variable values if required.

The execution in DWE Design Studio is intended for testing purposes. To promote them to production, you must deploy them through the use of the DWE Admin Console hosted on a WebSphere Application Server.

Control flows

A *control flow* is a container model that sequences one or more data flows and integrates other data processing rules and activities.

In Figure 6-38, there is a simple example that processes two data flows in sequence. If they fail, an e-mail is sent to an administrator.

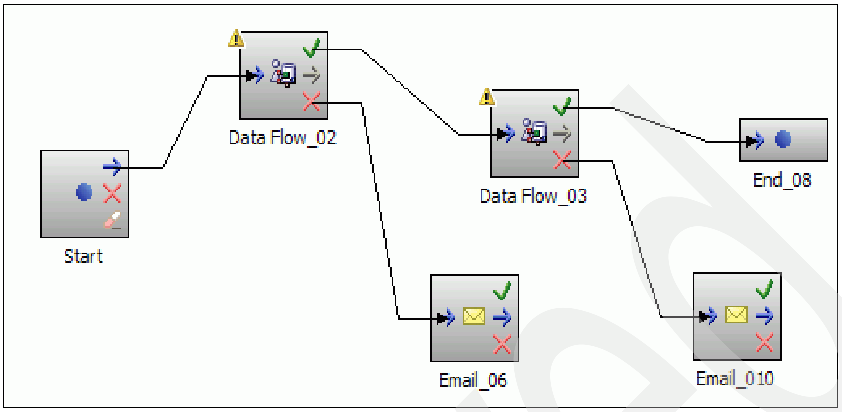


Figure 6-38 Simple control flow example

Control flows has several standard operators, such as:

- Start and end placeholders
- Mining flow
- Command (shell scripts, executables)
- Data Flow
- DataStage job sequence
- Email
- DataStage parallel job
- Iterator
- File wait

The operations palette is depicted in Figure 6-39 on page 177.

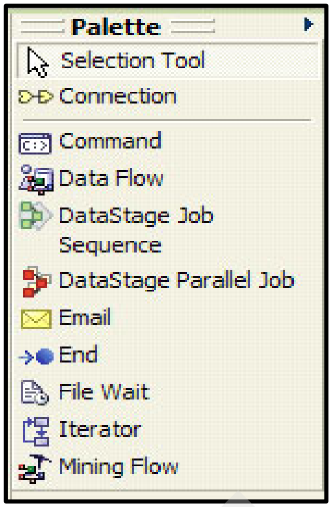


Figure 6-39 Control flow operators

Deployment

Deployment of SQLW applications execute based on the following steps:

1. Define required data sources and system resources in the WebSphere Application Server environment.
2. Enter the location of the zip file that was created by the preparation step in the Design Studio.
3. Link the data sources that the application needs to the data sources previously created.
4. Link the system resources (if any) that the application needs to the system resources previously created.
5. If your application contains variables that can be altered after the design and deployment preparation phases, you need to enter variable values or accept the current default values.

6.2 WebSphere Information Integration

An On Demand Business requires business processes, systems, and people to be fully integrated within your company and with partners, suppliers, and customers. Integration is at the heart of the On Demand Business.

For example, we have described six of the critical business integration initiatives addressed by IBM WebSphere Information Integration and have depicted them in Figure 6-40.

1. Master Data Management: Reliably synchronize all important business information dimensions, such as customers and products, across multiple systems.
2. Business Intelligence: Take the guesswork out of important decisions by consolidating trusted information in whatever form is needed, whenever it is needed.
3. Business Transformation: Transform companies into On Demand Businesses by isolating users and applications from the underlying information complexity.
4. Infrastructure Rationalization: Streamline corporate information access and reduce costs through an optimized information infrastructure.
5. Risk and Compliance: Deliver a dependable information management foundation to improve corporate visibility, ensure regulatory compliance, and lower operational risk.
6. Corporate Portals: Provide information on demand while isolating users from the complexities of multiple data sources and application interfaces.

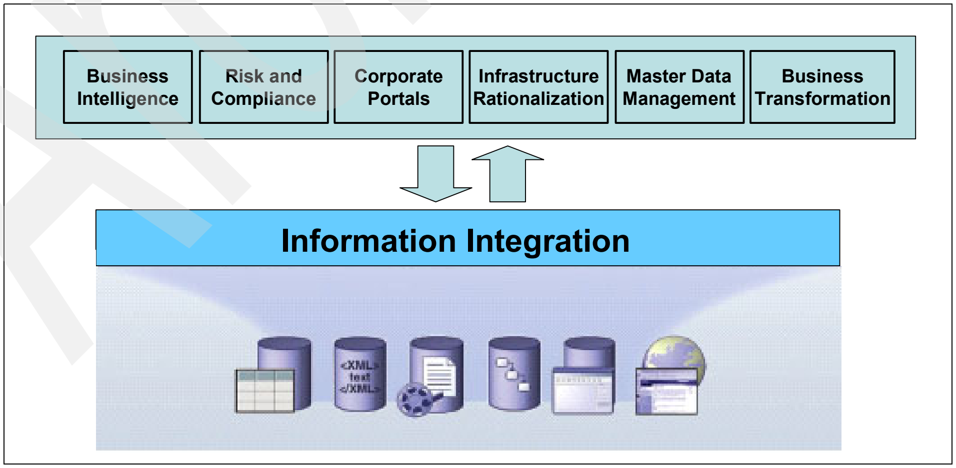


Figure 6-40 Business initiatives and Information Integration

As businesses replace manual processes with electronic processes and integrate processes across the enterprise, they must meet the challenge of finding, accessing, integrating, synchronizing, and sharing a wide variety of information. WebSphere Information Integration software gives companies real-time, integrated access to business information structured and unstructured, mainframe and distributed, and public and private across and beyond the enterprise.

Such information can reside in diverse source systems such as Oracle databases, SAP applications, Microsoft spreadsheets, flat files, the Web, and news groups and be distributed across a variety of operating environments such as Microsoft Windows, Linux, UNIX®, and IBM z/OS operating systems.

The software components that comprise WebSphere Information Integration are:

- WebSphere Information Integrator
- WebSphere DataStage
- WebSphere ProfileStage
- WebSphere QualityStage

6.2.1 WebSphere Information Integrator

WebSphere Information Integrator software offers a range of capabilities, such as enterprise federation, transformation, data placement (including replication and caching), and event publishing designed to meet varied integration requirements and easily integrate with industry-leading analytical tools, portal environments, packaged applications, application development environments, messaging-oriented middleware, service oriented architectures (SOAs), and business process software. With these capabilities, business intelligence and business integration applications can find and access diverse and distributed information as if it were a single source, regardless of where it resides. Changes to information can be monitored in order to notify individuals or to trigger business processes. Moreover, administrators can more easily distribute, consolidate, and synchronize information to facilitate application integration, maintain data warehouses, and support business continuity across complex, multiplatform, and multivendor IT environments.

A comprehensive information integration platform must provide five fundamental capabilities in order to deliver information across a full range of business requirements:

- The ability to connect to relevant sources whether structured or unstructured, mainframe or distributed, internal or external
- The ability to gain insight into the content, quality, and structure of data sources in order to completely understand data before it is integrated and proliferated throughout the enterprise
- The ability to standardize and cleanse the data so that companies gain access to authoritative and consistent views of any individual or business entity and its relationships across the extended enterprise
- The ability to effectively and efficiently collect, transform, and enrich high volumes of data from the original data source to the target
- The ability to federate information, enabling applications to access and integrate diverse data as if it were a single resource, regardless of where the information resides

These five fundamental capabilities are provided by the IBM WebSphere Information Integration platform. That platform integrates and transforms data to deliver information, providing breakthrough productivity, flexibility, and performance. These capabilities positioned on the IBM WebSphere Information Integration framework are depicted in Figure 6-41.

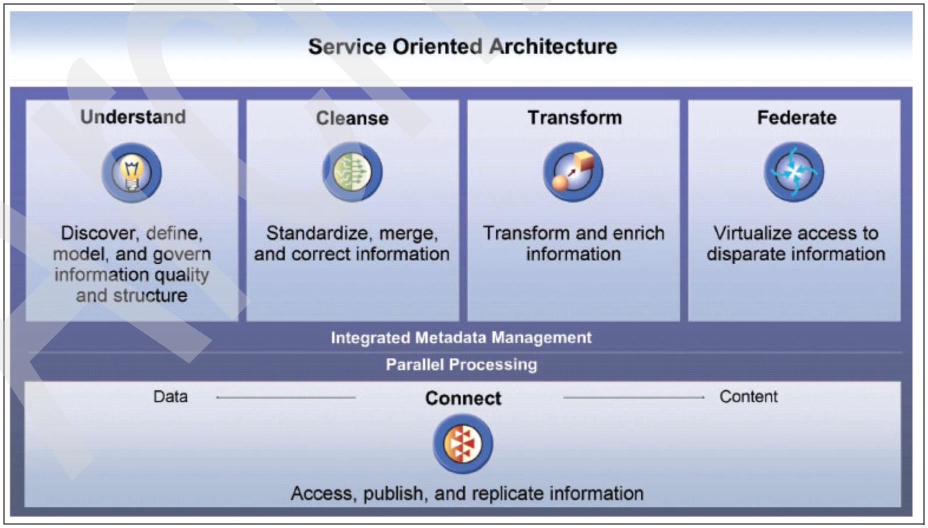


Figure 6-41 WebSphere Information Integration

The platform

The IBM WebSphere Information Integration platform:

- Delivers accessible, authoritative, consistent, timely, and complete information
- Provides leverage for businesses by allowing multiple types of business problems to be solved in an integrated manner, with a high degree of reuse and standardization
- Supports validation, access, and processing rules that can be reused across projects, leading to a high degree of consistency, control over data, and efficiency in IT projects, both in the initial development and over time as these rules need to be adjusted to meet changing business requirements

Understand

Understand and analyze information, including its meanings, relationships, and lineage. Businesses today deal with massive volumes of data often without much insight into the content, quality, and structure of that data. Complex business transactions from customers and partners plus operational information moving within the enterprise are often the basis on which key business decisions are made. These decisions are often undermined by the lack of insight and understanding of the data. WebSphere Information Integration solutions provide the automated data profiling, analysis, and monitoring capabilities to gather information about source data content, quality, and structure.

WebSphere Information Integration solutions provide:

- Table and column data-driven analysis and reporting to help identify missing, inaccurate, redundant, and inconsistent data
- Data quality monitoring to help maintain the health of data throughout its life cycle
- Automated discovery, and relationship and dependency analysis to establish the true metadata of the source systems
- The ability to define, annotate, and report on fields of business data

The product supporting these capabilities is IBM WebSphere ProfileStage.

WebSphere Information Integration shares a common metadata foundation, allowing metadata to be shared and tracked across products. This approach to integration development results in faster implementation times and better collaboration between IT and business users.

Cleanse

The combination of customer, partner, and operational information provides the foundation for key business decisions made across the enterprise. The more error-laden these data streams and the more prolific the data, the less confident the decision makers are in using this information to drive the business. Business information needs to be clean: identified, standardized, matched, reconciled, and free of redundancies to ensure quality and consistency. Data cleansing enables the establishment of a single, logically correct view of core business entities across the enterprise, the foundation for Master Data Management.

WebSphere Information Integration solutions provide:

- Standardization of source data fields, helping establish consistency in information
- Validation, certification, and enrichment of common data elements, using trusted data such as postal records for name and address information
- Matching together records across or within data sources, providing assurance that duplicate data is removed and enabling common entities from different sources to be identified and linked together
- The ability to allow a single information record to survive from the best information across sources for each unique entity, resulting in the creation of a single comprehensive and accurate view of information that spans across source systems

These functions can be applied to any type of business entity, including customers, products, suppliers, employees, and chart of accounts. They are vital to improving information quality and enabling a comprehensive view of information about your most important business assets.

The WebSphere Information Integration product supporting data cleansing is IBM WebSphere QualityStage.

Transform

Information today flows in, through, and out of business systems and processes like a living organism. Businesses need to tap into that data flow, regardless of its complexity, transform it by formatting it as required, and deliver it to the right targets or users at exactly the right time. Transformation technologies can increase productivity by meeting the most demanding information integration requirements for business intelligence, infrastructure rationalization, Master Data Management, regulatory compliance, data governance and other initiatives.

WebSphere Information Integration software provides high-volume transformation and movement functionality for complex data, for both batch and real-time processes. It supports integration of both System z data and data from distributed platforms. And it includes both client-based tools for design, administration, and operation and server-based data integration capabilities accessed through a common services layer.

WebSphere Information Integration solutions provide:

- High-volume, complex data transformation and movement functionality that can be used for stand-alone ETL or as a real-time data transformation engine for applications or processes
- Embeddable inline validation and transformation of complex data types, such as EDI, SWIFT, HIPAA, and other semi-structured data formats

The product supporting transformation is IBM WebSphere DataStage. WebSphere DataStage is described in detail in 6.2.2, “WebSphere DataStage” on page 188.

Federate

WebSphere Information Integrator V8.2 software provides federation for enterprise information integration. The *federation capability* refers to the ability to allow applications to access and integrate diverse data, mainframe and distributed, and public and private as if it were a single resource, regardless of where the information resides, while retaining the autonomy and integrity of the data sources.

WebSphere Information Integrator software has set the standard for federation with the following capabilities and characteristics:

- Transparency: Helps mask the differences, idiosyncrasies, and implementations of underlying information sources from the user, making the set of federated sources appear as a single system.
- Heterogeneous data access: Enables federated access to highly diverse types of data.
- Extensibility: Extends federated access to most data sources. Development and administration tools have been designed to minimize the effort required to integrate a new source, yet offer the flexibility to provide the information necessary to optimize query access.
- Rich functionality: Includes standard SQL-supported functions, compensation for missing functions in back-end data sources, and the ability to utilize source-unique capabilities and additional value-added capabilities beyond those natively delivered in the source systems.
- Information source autonomy: Enables federation of data sources with no impact to existing applications.
- Industry-leading performance: The SQL-based federation uses performance-oriented capabilities such as cost-based distributed query optimization, parallel and anticipatory execution, native source access, and transparent caching for great performance.

Overall, the WebSphere Information Integrator federation capability helps developers write less integration code, allowing access to more information. It also provides a reusable information asset interface that can be leveraged as part of future applications and initiatives.

Federation via SQL-based access paradigm

WebSphere Information Integrator federation, via the SQL-based access paradigm, provides access to the entire range of enterprise data sources either directly or via interoperability between other WebSphere Information Integrator offerings. Using SQL and standard open database connectivity (ODBC) and Java database connectivity (JDBC) APIs, WebSphere Information Integrator software fits neatly and transparently behind common analytical and reporting tools, development environments, portals, ETL tools, and other standard IT infrastructure components. SQL requests can be quickly and programmatically converted to Web services in an SOA. Alternatively, result sets can be programmatically converted into an XML document, validated, and published with a single SQL request.

Data sources accessible via SQL-based access in WebSphere Information Integrator V8.2 software include:

- Relational databases: IBM DB2 Universal Database (UDB), IBM Informix Dynamic Server, Informix Extended Parallel Server, Microsoft SQL Server, Oracle, Sybase SQL Server, Sybase Adaptive Server Enterprises, Teradata, and ODBC sources
- Mainframe databases: VSAM, IMS, Software AG Adabas and Computer Associates CA-Datacom and CA-IDMS via separate purchase of WebSphere Information Integrator Classic Federation for z/OS
- Packaged applications: SAP, PeopleSoft, and Siebel via separate purchase of IBM WebSphere Business Integration Adapters
- Life sciences sources: Kyoto Encyclopedia of Genes and Genomes (KEGG) and data sources accessible by Entrez, BLAST, HMMER (including support for the HMMSEARCH tool), and BioRS
- Other: WebSphere MQ message queues, Microsoft Excel spreadsheets, table-structured flat files, XML documents, data sources accessible via OLEDB and Web services, including complex XML results such as those providing access to legacy applications, or other data integration tools
- C++ and Java developer toolkits to add access to other sources

The WebSphere Information Integration products supporting federation are:

- IBM WebSphere Information Integrator Standard and Advanced Editions
- IBM WebSphere Information Integrator Classic Federation for z/OS

Connect

With businesses more distributed, consolidating, synchronizing, and distributing data across disparate databases is a core business requirement. The WebSphere Information Integration portfolio meets these business demands, enabling businesses to connect to all their information. Solutions provide:

- Direct, native access to relevant sources bundled with each product for both mainframe and distributed computing environments
- Consolidation, synchronization, and distribution across disparate sources
- Support for a wide range of information sources, such as databases, files, and packaged applications
- Changed data capture and event-based publishing of data

WebSphere Information Integration connectivity products can be used stand-alone to support specific application requirements or in conjunction with the other products in the platform to provide integrated composite solutions.

Replication

WebSphere Information Integrator V8.2 software offers replication that helps administrators distribute, consolidate, and synchronize information across complex, multiplatform, and multivendor IT environments. This software provides both queue-based and SQL-based replication architectures that present distinct benefits for various business needs.

Replication is used in a variety of contexts:

- Facilitate application integration. Whether point-to-point or distribution and consolidation topologies are required, it lets you manage data consistency between different application domains. For example, a retailer might replicate orders from showrooms to the production server and the latest inventory from the production server to showroom systems.
- Maintain data warehouses. Helps you utilize current information, capturing changes from transaction databases and replicating them into operational data stores, data warehouses, or data marts to facilitate real-time business intelligence.
- Support business continuity. Can maintain synchronization for local or remote backup systems in either a standby or active mode.

Administrators can use a wizard-driven graphical user interface (GUI), command-line processor, and script-driven processes to configure the variety of topologies, latency, and consistency characteristics for both queue-based and SQL-based replication architectures. Integrated monitoring and reconciliation tools make it easy for administrators to react to problems and proactively maintain the health of the environment.

Queue-based replication

For DB2 Universal Database data sources and targets, queue-based replication architecture offers low latency and high-volume replication with managed conflict detection and resolution. Queue-based replication is designed to support business continuity, workload distribution, and application integration scenarios.

Committed changes are captured from the database log and placed onto a WebSphere MQ message queue. A sophisticated apply process engine determines transaction dependencies and replays transactions on target systems to maximize parallelism and minimize latency. A set of conflict detection and resolution algorithms identifies conflicting updates from peer systems, allowing backup systems to work productively, and application workloads to be distributed across multiple servers. In addition, data can be filtered so that only the data of interest is replicated, stored procedures can be invoked to facilitate transformations, and programmatic and high-performance load options can perform simultaneous loading and replicating.

SQL-based replication

For replication among databases from multiple vendors, WebSphere Information Integrator software uses an SQL-based replication architecture that maximizes flexibility in managing scheduling, transformation, and distribution topologies for populating data warehouses or data marts, maintaining data consistency between disparate applications or efficiently managing distribution and consolidation scenarios among headquarters and branch or retail configurations.

In SQL replication, changes are captured with either a log-based or trigger-based mechanism and inserted into a relational staging table. An apply process asynchronously reads the changes from the staging table and handles the updates to the target systems.

Data can be distributed or consolidated via the replication server, and the data movement can be continuous, event-driven, or automated on a specific schedule or at designated intervals. As with queue-based replication, data can be filtered so that only the data of interest is replicated. Moreover, standard SQL expressions and stored procedures can be invoked to facilitate transformations, and management of the data movement can be table-at-a-time for batch warehouse loading or transaction-consistent to maintain continuous online availability.

With SQL-based replication, you can replicate data between mixed relational data sources:

- Supported as sources and targets: DB2 Universal Database, Informix Dynamic Server, Microsoft SQL Server, Oracle, Sybase SQL Server, and Sybase Adaptive Server Enterprise
- Supported as targets: Informix Extended Parallel Server and Teradata

Event publishing

WebSphere Information Integrator V8.2 software links data events with business processes, capturing database changes from mainframe data sources by reading the recovery log, formatting the changes into XML messages, and then publishing them to WebSphere MQ. Any application or service that integrates with WebSphere MQ directly or supports Java Message Service (JMS) can asynchronously receive data changes as they occur. For example, using the event publishing capability, WebSphere Business Integration software can receive changes from a DB2 UDB database as they occur and can programmatically update an SAP application. Alternatively, a JMS-aware application or service within any Java 2 Platform Enterprise Edition (J2EE) server (for example, WebSphere Application Server) could receive those same changes and perform additional processing or integration. In addition, WebSphere Information Integrator event publishing solutions offer message formatting flexibility and allow changes to be filtered in multiple ways.

WebSphere Information Integrator event publishing solutions enable database events to initiate business processes. For example, a reduction in an inventory value could be used to drive a product restocking workflow, or the addition of a new customer could initiate a welcome e-mail, credit verification, and an accounting update. This creates an application-independent, loosely coupled integration that is adaptable to changing application environments. For instance, while multiple applications might affect the value of the inventory level, a single point of integration, the data items themselves, is driving the workflow. Changes to the applications that affect the inventory level can be made with no impact on the event-driven integration.

Businesses can realize a faster time-to-market based on integration that captures the event in a single location, is more stable, and is easier to maintain than integrations that attempt to monitor multiple applications for events.

Additionally, event publishing can deliver changes to ETL tools, custom-built processes for updating operational data stores or data warehouses that minimize bandwidth requirements and keep target databases more closely in sync. Thus, businesses can utilize current information for tactical and operational decision making.

Event publishing is available from DB2 UDB for z/OS; DB2 UDB for Linux, UNIX and Windows; IMS; VSAM; and Computer Associates CA-IDMS sources.

The WebSphere Information Integration products supporting connect capabilities are:

- IBM WebSphere Information Integrator Replication Editions
- IBM WebSphere Information Integrator Event Publisher Editions
- IBM WebSphere Information Integrator Standard Edition
- IBM WebSphere Information Integrator Classic Federation for z/OS

6.2.2 WebSphere DataStage

IBM WebSphere DataStage, a core component of the IBM WebSphere Information Integration, enables you to tightly integrate enterprise information, despite having many sources or targets and short time frames. Whether you are building an enterprise data warehouse to support the information needs of the entire company, building a real-time data warehouse, or integrating source systems to support enterprise applications such as CRM, SCM, and ERP, WebSphere DataStage supports the enterprise data integration initiatives.

WebSphere DataStage supports the collection, integration, and transformation of high volumes of data, with data structures ranging from simple to highly complex.

WebSphere DataStage can operate in real-time, capturing messages or extracting data at a moment's notice on the same platform that also integrates bulk data.

WebSphere DataStage delivers four core capabilities, all of which are necessary for data transformation within enterprise data integration projects:

- Connectivity to a wide range of mainframe, legacy, and enterprise applications, databases, and external information sources to ensure that critical enterprise data assets can be used
- Intrinsic, prebuilt library of 300 functions to reduce development time and learning curves, increasing data accuracy and reliability and providing reliable documentation that lowers maintenance costs
- Maximum throughput from any hardware investment used in the completion of bulk tasks within the smallest batch windows, and the highest volumes of continuous, event-based transformations using a single high-performance parallel processing architecture
- Enterprise-class capabilities for development, deployment, and maintenance with no hand-coding required; and high-availability platform support to reduce ongoing administration and implementation risk

WebSphere DataStage is a component of WebSphere Information Integration and is integrated with data profiling, data quality, and cleansing products for scalable enterprise data integration solutions.

WebSphere DataStage can operate in real time, capturing messages or extracting data on the same platform that also integrates bulk data. This allows you to respond to your data integration needs on demand. WebSphere DataStage transformation topology is depicted in Figure 6-42.

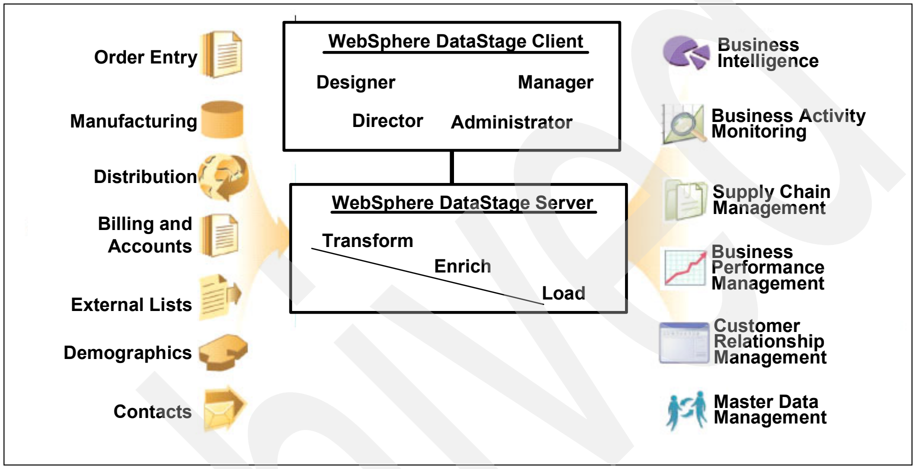


Figure 6-42 Transformation topology

WebSphere DataStage supports a large number of heterogeneous data sources and targets in a single job, including:

- Text files
- Complex XML data structures
- Enterprise application systems including SAP, Siebel, Oracle, and PeopleSoft
- Almost any database, including partitioned databases such as Oracle, IBM DB2 Universal Database (with and without the Data Partitioning Feature), IBM Informix, Sybase, Teradata, and Microsoft SQL Server
- Web services
- SAS
- Messaging and enterprise application integration products including WebSphere MQ and SeeBeyond

Development environment

WebSphere DataStage employs a *work as you think* design metaphor. Developers use a top-down data-flow model of application programming and execution, which allows them to create a visual sequential data flow. A graphical palette helps developers diagram the flow of data through their environment via GUI-driven drag-and-drop design components. Developers also benefit from scripting language, debugging capabilities, and an open application programming interface (API) for leveraging external code. The WebSphere DataStage Designer tool is depicted in Figure 6-43.

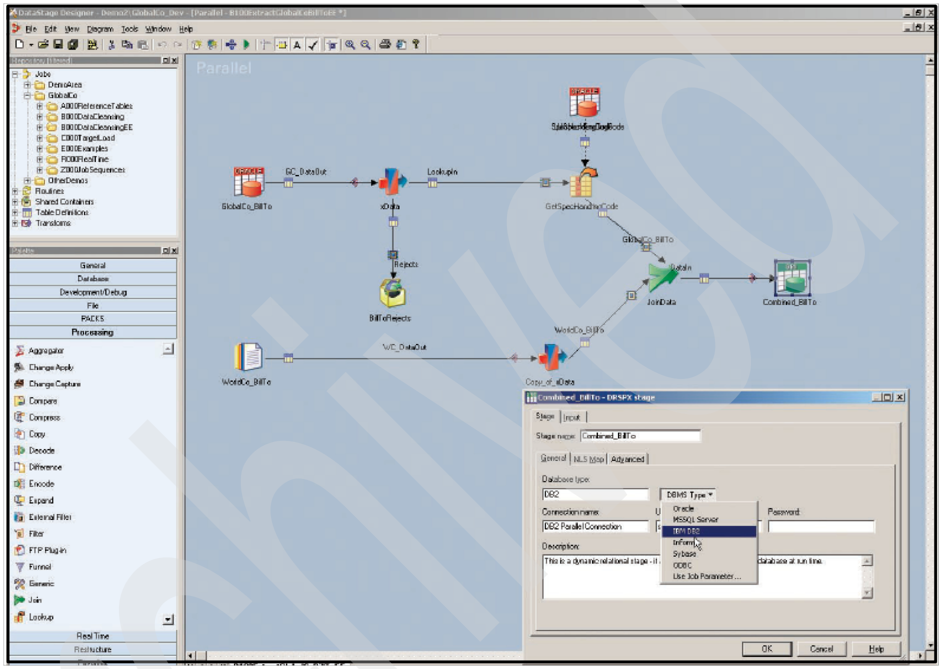


Figure 6-43 WebSphere DataStage Designer

6.2.3 WebSphere ProfileStage

WebSphere ProfileStage allows users to integrate multiple disparate systems by providing a complete understanding of the metadata and by discovering dependencies within and across tables and databases. Because the metadata is based upon the actual source data, accuracy is nearly 100%, reducing the project risk by uncovering integration issues before development begins.

WebSphere ProfileStage brings automation to the critical and fundamental tasks of data source analysis, expediting comprehensive data analysis, reducing the time-to-market, and minimizing overall costs and resources for critical data integration projects. It profiles source data by analyzing column values and structures and provides target database recommendations, such as primary keys, foreign keys, and table normalizations. Armed with this information, it builds a model of the data to facilitate the source-to-target mapping and automatically generates integration jobs.

Some of the functions and features of WebSphere ProfileStage are:

- Analyzes and profiles source and target systems to enable discovery and documentation of data anomalies
- Validates the content, quality, and structure of your data from disparate systems without programming
- Enables metadata exchange within the integration platform
- Provides a single and open repository for ease of maintenance and reporting

No assumptions are made about the content of the data. The user supplies a description of the record layouts. Then WebSphere ProfileStage reads the source data and automatically analyzes and profiles the data so that the properties of the data (defined by the metadata) are generated without error. The properties include the tables, columns, probable keys, and interrelationships among the data. Once these properties are known and verified, WebSphere ProfileStage automatically generates a normalized target database schema.

You specify the business intelligence reports and source data to target database transformations as part of the construction of this target database. After the source data is understood, it must be transformed into a relational database. This process is automated by ProfileStage, yielding a proposal for the target database that can be edited to get the best possible results.

The following is a description of the process and major components for profiling:

- Column Analysis: Here we examine all values for the same column to infer the column definition and other properties such as domain values, statistical measures, and min/max values. During Column Analysis, each available column of each table of source data is individually examined in-depth. It is here that many properties of the data are observed and recorded, such as minimum, maximum, and average length, precision and scale for numeric values, basic data types encountered including different date and time formats, minimum, maximum and average numeric values, count of empty values, NULL values, and non-NULL/empty values, and count of distinct values or cardinality.
- Table Analysis: This is the process of examining a random data sample selected from the data values for all columns of a table in order to compute the functional dependencies for this table. The purpose is to find associations between different columns in the same table. A functional dependency exists in a table if one set of columns is dependent on another set of columns. Each functional dependency has two components:

– Determinant: A set of columns in the same table that composes the determinant. That is, the set of columns that determines the dependency. A determinant can consist of one or more columns.

– Dependent Column: One column from the same table that is dependent on the determinant. A column is said to be dependent if, for a given value of the determinant, the value of the dependent column is always the same.

- Primary Key Analysis: The process of identifying all candidate keys for one or more tables. The goal is to detect a column, or set of columns, which might be best suited as the primary key for each table. This analysis step must be completed before subsequent steps, such as Cross Table Analysis, can be performed. Normally, Primary Key Analysis uses results from Table Analysis. Table Analysis identifies the dependencies among the columns of a table and records each as an aggregate dependency.

  Each row in the Aggregate Dependency table represents a single dependency for a given table. Each dependency has two components: a single column or set of columns (in the same table) that makes up the determinant, and a set of columns (also in the same table) that are dependent upon the determinant. A set of columns is dependent on a determinant if, for a given value of the determinant, the value of the dependent columns is always the same. As you would then expect, a primary key determines all the values for the rest of the columns in the table. During Primary Key Analysis, one or more of the aggregate dependencies become candidate keys. Subsequently, one candidate key must be confirmed by the user as the primary key.

- Cross-Table Analysis: This is the process of comparing all columns in each selected table against all columns in the other selected tables. The goal is to detect columns that share a common domain. If a pair of columns is found to share a common domain, then this might indicate the existence of a foreign key relationship between the two tables, or simply redundant data. These possibilities are examined during the subsequent Relationship Analysis step.

  Each row in the Domains Compared table represents a pair of columns whose domains have been compared during Cross-Table Analysis. A domain comparison is a bidirectional process, which might conclude that one column's domain is contained within that of the other column, or vice versa. Each row in the Common Domains table represents the fact that one column (the base column) shares a common domain with another column (the paired column) in a different table. The common domain is noted only from the perspective of the base column; it makes no representation of whether or not a common domain also exists in the other direction.

- Normalization: Involves computing a third normal form relational model for the target database. The user interface provides a Normalization Wizard that guides the user through the process of normalizing the target database model. The information gained through the analysis phases is used to aid the user in making intelligent decisions in the construction of the target data model. When WebSphere ProfileStage spots a candidate for normalization, the user is presented with a proposed normalization. The user can accept the proposed normalization, reject the normalization, or modify the model.

- Reporting and Data Definition Language (DDL) Generation: The profiling reports describe in detail the information gained from the profiling steps. These reports can be used as the basis for estimating the scope of the project, for obtaining signoff from users and stakeholders, and investigating the true composition of the source data. The reports can be output to the user's window, a printer, a file, e-mail, Word, HTML, and a variety of other formats.

  The data model constructed can be exported to popular data modeling tools and in a variety of formats. The user can then examine the data model in a variety of combinations. If after examining the data model, the user determines that changes in the target schema are necessary, values can be adjusted in the Normalization Wizard or in the analysis phases. New or revised models can be loaded into the WebSphere ProfileStage Metadata Repository and integrated into the project. WebSphere ProfileStage supports generation of SQL in a variety of dialects, including SQL Server, ANSI SQL, and Oracle. The DDL can also be generated in XML format.

- Support for ETL Tools: Once the mappings have been confirmed, creating the ETL jobs to perform the creation of the target database is merely the push of a button. This approach also supports mapping from sources to predefined targets with a drag-and-drop interface.

  The code for WebSphere DataStage job transforms is automatically generated. An example of this is depicted in Figure 6-44 on page 194. Here, a non-normalized source database is converted into a fully normalized target database. No programmer time was necessary to build the WebSphere DataStage jobs for these basic transformations. Because the WebSphere ProfileStage approach derives the data model for the target database from the information stored in the WebSphere ProfileStage Metadata Repository, the source-to-target mappings are automatically computed.

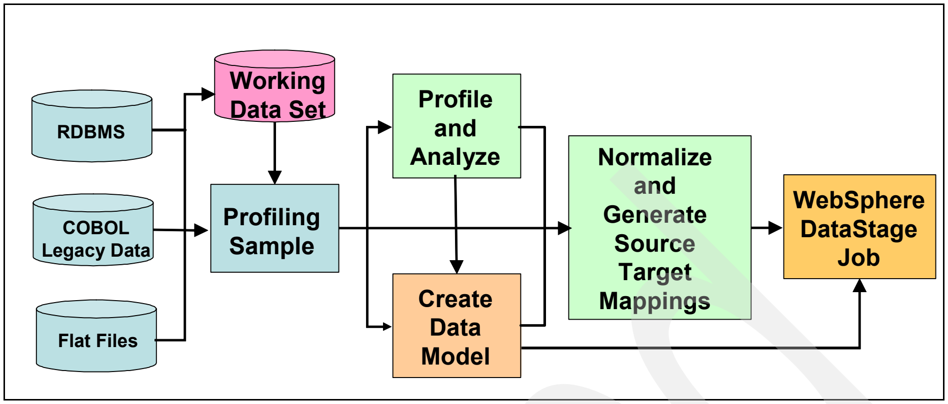


Figure 6-44 Data profiling for data transformation

6.2.4 WebSphere QualityStage

WebSphere QualityStage helps your strategic systems deliver accurate, complete information to business users across the enterprise. Through an easy-to-use GUI and capabilities that can be customized to your organization's business rules, WebSphere QualityStage provides control over international names and addresses and related data such as phone numbers, birth dates, e-mail addresses, and other descriptive comment fields and discovers relationships among them in enterprise and Internet environments, in batch and real time.

It analyzes data at the character level and uncovers anomalies and buried data prior to transforming it for database loading or transaction processing. Data from disparate sources is standardized into fixed fields using business driven rules to assign the correct semantic meaning to input data in order to facilitate matching.

Duplication and data relationships can be detected despite anomalous, inconsistent, and missing data values. There is a unique statistical matching engine that assesses the probability that two or more sets of data values refer to the same business entity - providing the most accurate match results available.

Once a match is confirmed, linking keys are constructed so users can complete a transaction or load a target system with true entity integrity and view related data as information. As a result, companies gain access to accurate, consistent, and consolidated views of any individual or business entity and its relationships across the enterprise.

An example of a typical quality process is depicted in Figure 6-45 on page 195.

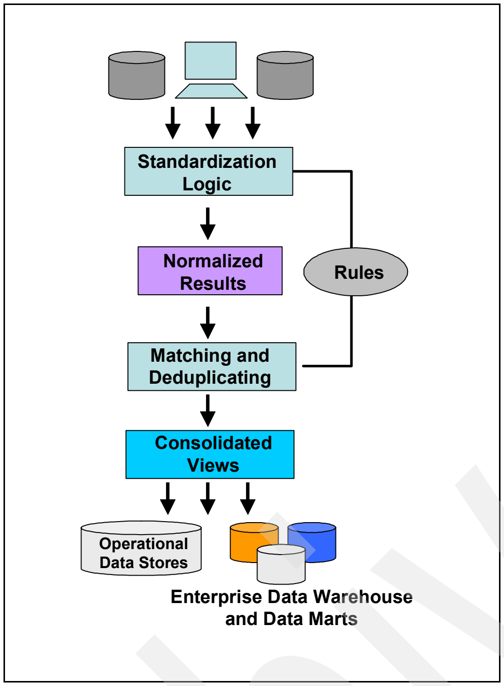


Figure 6-45 QualityStage Process

6.3 WebSphere Portal

Portals serve as a simple, unified access point to Web applications, but also do more. They provide valuable functions such as security, search, and collaboration. A portal delivers integrated content and applications, plus a unified, collaborative workplace. Indeed, portals are the next-generation desktop, delivering e-business applications over the Web to all types of client devices.

A complete portal solution should provide a convenient access to everything needed to get tasks done anytime, anywhere, and in a secure manner. Portals are the key to providing a personalized, relevant Web experience, enabling users to find what they need in a highly interactive and personal way. That is, portals provide the tools and user interface to access information and applications, and to manage the selection and personalization of content.

WebSphere Portal provides an extensible framework for interacting with enterprise applications, content, people, and processes. Self-service features allow users to personalize and organize their own view of the portal, to manage their own profiles, and to publish and share documents with their colleagues.

WebSphere Portal software is an open standards-based framework supporting a range of options across databases, directories, platforms, and security, with portlets serving as a key component. The term *portlet* refers to a small portal application, usually depicted as a small box in the Web page. Portlets are reusable components that provide access to enterprise applications, Web-based content, host and data systems, content-management systems, process-driven applications, and other resources. Web pages, Web services, applications and syndicated content feeds can be accessed through portlets. WebSphere Portal software includes pre-integrated portlets, cross-portlet integration for all application types, tools that enable you to create new portlets, and the ability to construct composite applications to manage business processes and transactions spanning multiple enterprise systems. As a result, WebSphere Portal helps organizations move beyond fragmented application silos while hiding the complexity of the IT infrastructure.

WebSphere Portal uses the advantages provided by the IBM WebSphere software platform. That platform delivers standards-based integration and infrastructure software to maximize business flexibility and responsiveness. It is built with services to support scalable, reliable Web applications, with components and technologies that enable you to add extensions to new applications and processes, and provide integrated collaborative services.

WebSphere Portal software provides an extensible framework for interacting with enterprise applications, content, people, and business processes. It includes IBM WebSphere Application Server and WebSphere Business Integration Server Foundation components to support scalable Web application server and business-process integration solutions managed from the portal framework. WebSphere Portal self-service features enable users to personalize and organize their own view of the portal, to manage their own profile, and to publish and share documents with their colleagues.

6.3.1 Architecture

By integrating key IBM products and technologies and providing application programming interfaces (APIs) and extension points for ISVs and clients to extend and customize their environments, WebSphere Portal represents the defacto standard for an On Demand Business architecture and is depicted in Figure 6-46 on page 197.

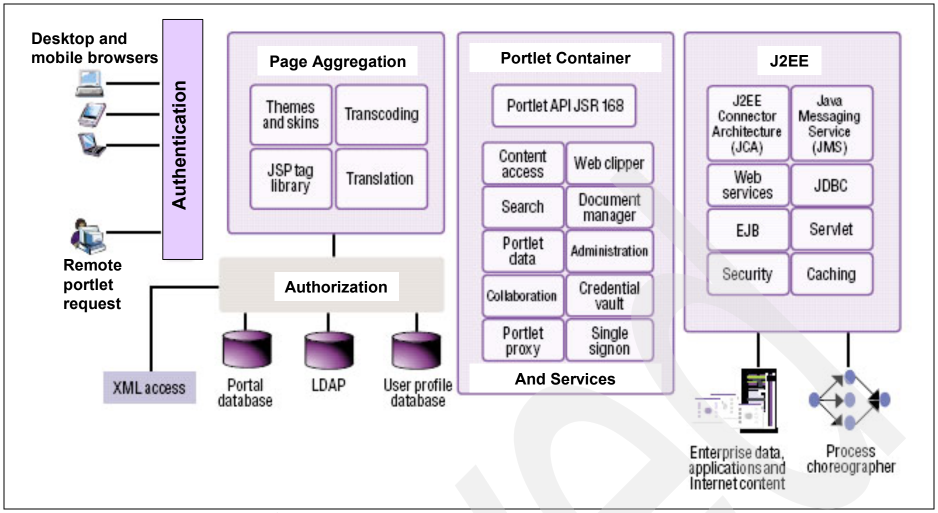


Figure 6-46 WebSphere Portal architecture

An On Demand Business is an enterprise where business processes, integrated end-to-end across the company and with key partners, suppliers, and customers, can respond with speed to customer demand, market opportunity, or external threat. More than operational efficiency, On Demand Business is about building a dynamic infrastructure that allows you to integrate, modify, and leverage existing applications and processes cost-effectively.

As Web-based applications enter the era of On Demand Business, WebSphere Portal includes delegated administration, cascading page layouts, and portal federation through Web services. It also includes support for standards, advanced portlet application concepts, process integration, task management, and search services.

IBM WebSphere Portal is available in two editions, Enable and Extend. Their models and descriptions are summarized in Table 6-4 on page 198. Each is designed to provide the infrastructure for specific needs to build and deploy scalable portals. These offerings share a common framework (the portal server) and might share certain products and services. The portal server provides common services, such as application connectivity, integration, administration, and presentation that are required across portal environments.

Table 6-4 WebSphere Portal Enable and Extend

Enable	Extend	Component	Description
Yes	Yes	IBM WebSphere Portal server	Provides presentation, user management, security, and other services for constructing a portal.
Yes	Yes	IBM WebSphere Portal personalization engine	Includes advanced personalization technologies to target Web content to meet user needs and preferences.
Yes	Yes	IBM Workplace Web Content Management	Provides tools for authoring, personalizing, and publishing content and documents contributed to the portal.
Yes	Yes	IBM WebSphere Portal Document Manager	Centralizes document storage, organization, and version-management services.
Yes	Yes	Productivity components	Offers an inline view and edit capabilities designed to support rich text, spreadsheet, and presentation content.
Yes	Yes	IBM Tivoli Directory Server	Provides a Lightweight Directory Access Protocol (LDAP) server.
Yes	Yes	IBM WebSphere Translation Server	Delivers translation services that enable Web content, e-mail, and chat content conversions across languages.
Yes	Yes	IBM Rational Application Developer	Provides professional developer tools to create, test, debug, and deploy portlets, servlets, and other assets related to portals and Web applications.
Yes	Yes	IBM Lotus Collaboration Center	Includes preconfigured pages and portlets to deliver ready to use portlet access to IBM Lotus Domino and extended products, including IBM Lotus Notes and Domino mail and applications, IBM Lotus Domino Document Manager, IBM Lotus Instant Messaging (Sametime), IBM Lotus Team Workplace (QuickPlace), and IBM Lotus Web Conferencing Server.
Yes	Yes	Collaborative components	Includes Java application programming interfaces (APIs) that provide the building blocks for integrating the functionality of Domino, Lotus Instant Messaging, and Lotus Team Workplace into portals and portlets.

Enable	Extend	Component	Description
No	Yes	IBM Lotus Extended Search	Delivers parallel, distributed, heterogeneous searching capabilities across Lotus Notes and Domino databases, Microsoft sources, relational database management system (RDBMS) data stores, Web search sites, and other sources.
No	Yes	IBM Tivoli Web Site Analyzer	Analyzes Web-site usage logs to reveal information that you can use to improve your portal to provide better user experiences.
No	Yes	Lotus Instant Messaging	Offers instant messaging, presence awareness, and Web-conferencing services.
No	Yes	Lotus Team Workplace	Provides a Web-based solution to create team workspaces for collaboration. Features include discussions, document collaborations, and the ability to orchestrate plans, tasks, and resources.

As part of the WebSphere Portal family in place to support small and medium business, WebSphere Portal Express and WebSphere Portal Express Plus make it easier to share information and enhance team productivity by enabling employees to view, search, create, convert, and edit basic documents, spreadsheets, and presentations.

6.3.2 Portlets

Any particular portlet is developed, deployed, managed, and displayed independent of other portlets. Personalized portal pages can be created by choosing and arranging portlets, resulting in Web pages that present content tailored for individuals, teams, divisions, and organizations. An example is depicted in Figure 6-47 on page 200.

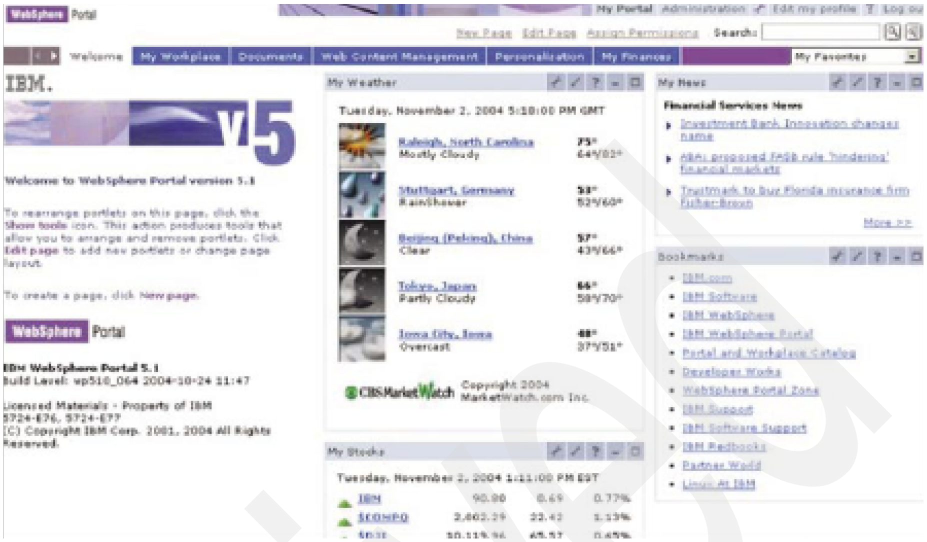


Figure 6-47 WebSphere Portal Server and servlets main window

The portal server includes a rich set of standard portlets for storing and sharing documents, displaying syndicated content, and performing XML transformation. It also includes portlets for accessing existing Web pages and data applications, Lotus Notes and Microsoft Exchange productivity applications, Lotus Instant Messaging, and Lotus Team Workplace applications.

Portlet applications

Portlets are more than simple views of existing Web content. A portlet is a complete application, following a standard model-view-controller (MVC) design. Portlets have multiple states and view modes, as well as event and messaging capabilities. Portlets run inside the portlet container of a portal server, similar to a servlet running on an application server. The portlet container provides a runtime environment in which portlets are instantiated, used, and, finally, destroyed. Portlets rely on the portal infrastructure to access user-profile information, participate in window and action events, communicate with other portlets, access remote content, search for credentials, and store persistent data.

Generally, you can administer portlets more dynamically than servlets. For example, you can install or remove portlet applications consisting of several portlets while a server is running. You can also change the settings and access rights of a portlet while the portal is running, even in a production environment.

Portlet modes

Portlet modes allow a portlet to display a different user interface, depending on the task required of the portlet. A portlet has several modes of display, which can be invoked by icons on the portlet title bar. These modes include view, help, edit, and configure.

A portlet is initially displayed in its view mode. As the user interacts with the portlet, it might display a sequence of view states, such as forms and responses, error messages, and other application-specific states. The help mode provides user assistance about the portlet. Edit mode provides a page for users to change the portlet settings. For example, a weather portlet might include an edit page so that users can specify their location. Users must log in to the portal to access edit mode. If configure mode is supported by a portlet, it provides a page for portal administrators to configure portlet settings that are shared by instances of that portlet.

Each portlet mode can be displayed in normal, maximized, or minimized states. When a portlet is maximized, it is displayed in the entire body of the portal page, replacing the view of other portlets. When a portlet is minimized, only the portlet title bar is displayed on the portal page.

The portlet API

Portlets are a special subclass of the *HttpServlet* class with properties that allow them to easily plug into and run in the portal server. They are assembled into a larger portal page with multiple occurrences of the same portlet displaying different data for each user. The portlet API provides standard interfaces for portlet functions. It defines a common base class and interfaces for portlets to cleanly separate the portlet from the portal infrastructure. For the most part, the portlet API is an extension of the servlet API, except that it restricts certain functions to a subset that makes sense for portlets running in the context of a portal. For example, unlike servlets, portlets cannot send errors or redirect messages as a response. This can be performed only by the portal itself, which controls the overall response page.

The markup fragments that portlets produce can contain links, actions, and other content. The portlet API defines URL rewriting methods that allow portlets to transparently create links without needing to know how URLs are structured in the particular portal.

Portlet performance

Because portlets are essentially servlets, similar reentrance and performance considerations apply to both. A single portlet instance (that is, a single instance of the portlet Java class) is shared among all requesters. A limited number of threads can process portlets and servlets, so each portlet must do its job as quickly as possible to optimize response time for the whole page. Just as with servlet programming, you should consider optimizations such as limiting the use of synchronized methods, limiting the use of expensive string operations, avoiding long-running loops, and minimizing the number of objects created. You can also optimize response times by using Java Server Pages (JSP) to render the portlet views. In general, views created with JSP are faster than views created with Extensible Stylesheet Language (XSL).

Usually, several portlets are invoked in the course of handling a single request, each one appending its content to the overall page. Some portlets can be rendered in parallel, so that the portal server assembles all the markup fragments when all the portlets finish or time out. This improves the performance of portlets that access remote data by HTTP or Simple Object Access Protocol (SOAP) requests. However, not all portlets are thread-safe. For example, portlets that access protected resources cannot run in parallel. The portlet deployment descriptor indicates whether the portlet is considered thread-safe. Portlets that are not thread-safe are rendered sequentially.

Portlet output can also be cached. Caching policies are configured in the portlet deployment descriptor. You can include an expiration time and whether the portlet markup can be shared among users or is user-specific.

Standards

As portals continue to evolve into the new desktop and integration standard, IBM directs efforts to standardize the APIs between portals and other applications and often assumes lead technical positions within many standards organizations. In particular, the Java Community Process (JCP) and the Organization for the Advancement of Structured Information Standards (OASIS) work cooperatively to standardize the Java and XML technology needed to link portals to disparate applications. Some of these standards include or will include:

- Java Specification Request (JSR) 168, a specification from JCP that addresses the requirements of aggregation, personalization, presentation, and security for portlets running in a portal environment. JSR 168 is co-led by IBM and Sun Microsystems, Inc., and is designed to facilitate interoperability between local portlets and portal servers. WebSphere Portal includes a portlet runtime environment with a JSR 168 portlet container that supports operation of portlets developed according to the Java Portlet Specification defined by JSR 168.
- JSR 170, a proposed specification from JCP designed to implement a standard meta-model definition and an API for bidirectional, granular access to content repositories. This should result in platform-independent methods to interact across content-management solutions and associated services including versioning, search, content categorization, access control, and event monitoring. IBM is a participant in the expert-member group defining

the JSR 170 content repository for Java technology APIs. When JSR 170 is published, IBM plans to support this standard across its content-management offerings, and will use the JSR 170 repository to store all portal content.

- OASIS Web Services for Remote Portals (WSRP), an XML and Web services standard that enables the interoperability of visual, user-facing services with portals or other Web applications. The OASIS WSRP standard simplifies the integration of remote applications and content into portals by defining a Web service interface for interactive presentation-oriented Web services. The producer part of the WSRP implementation provides an entry point into the producer portal, enabling the portal to provide portlet applications or single portlets as WSRP services. A WSRP consumer is a portal that wants to integrate WSRP services and consume them as remote portlets from portals that provide them. As a result, using WSRP makes integrating content and applications into portals easier, eliminating the requirement for custom coding or the use of a variety of protocols and interfaces to adapt Web services for use in their portal implementation. WebSphere Portal, Version 5.1 includes support for producer and consumer WSRP services.
- Struts, a Jakarta open-source project that provides a framework based on the MVC pattern. It enables developers to efficiently separate an application's business logic from its presentation. Struts enforces sequences of pages and actions and provides form-validation functions. WebSphere Portal, Version 5.1 includes support for the Struts, Version 1.1 framework to build portlets. To work in portlets, you must observe specific technical details when using Struts. For example, when used in portlets, a Struts action should not write to the response object, and should not create header elements, such as HEAD and BODY tags. The Struts application must be packaged with several replacement Java Archive (JAR) files that help ensure that URLs, forward actions, and include actions run correctly in a portal environment.

Portlet cooperation

The portal server provides a mechanism for portlets to communicate with each other, exchanging data or other messages. In a production portal, you can use portlet communication to copy common data between portlets. This capability saves time by minimizing the need for redundant typing by the user and makes the portal easier to use. For example, one portlet might display information about accounts while a second portlet displays information about transactions that have occurred for one of the accounts over the past 30 days. To do this, the transactions portlet needs to obtain the corresponding account information when it displays the transaction details. This action is accomplished by communication between the two portlets, using portlet actions and portlet messages. For example, the account portlet creates a portlet action and encodes it into the URL that is rendered for displaying transactions. When the link is clicked, the action listener is called, and a portlet message is launched to send the necessary data.

Programmatic messaging helps unify portlet applications that access different back-end applications. However, programmatic messaging is relatively static and requires planning and design work in advance. The portlets that are exchanging messages must already know about each other to make the interchange work. The next section discusses a more flexible means of portlet cooperation.

Brokered cooperation

Brokered cooperation enables independently developed portlets to exchange information. Portlets register their intent to cooperate with a broker, which facilitates the exchanges at runtime. The broker works by matching data types between the sources in one portlet and the actions of another portlet. When the types match, a transfer is possible, and the broker enables the user to trigger the transfer through a pop-up menu. The term *click-to-action* is used to describe this menu-driven, brokered data exchange as depicted in Figure 6-48.

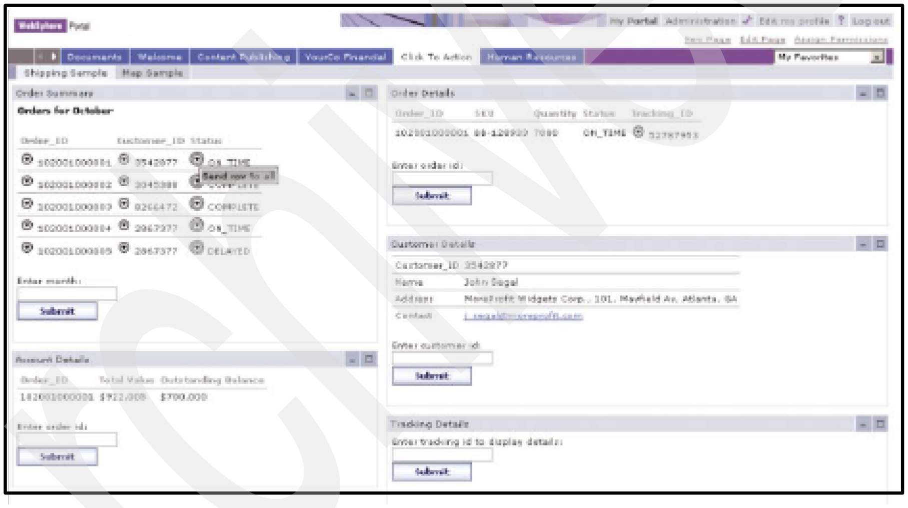


Figure 6-48 Brokered data exchange: Click-to-action

The objective of click-to-action portlets is to increase the productivity of portal users working with multiple portlets by easily enabling them to send information from one portlet to another. For example, users can click information that is displayed in one portlet and transfer that information to another portlet. The portlet receiving the information processes it and updates the display. The click-to-action capability programmatically matches portlet information sources and possible actions based on their data-type compatibility. Click-to-action does not rely on drag-and-drop or other nonstandard browser features. It also offers the unique advantage of being able to work in different browsers, which makes it more accessible to users.

An extension of this idea, *cooperative portlets*, is also supported by WebSphere Portal. Using cooperative-portlet capabilities, administrators can pre-wire portlets so that they exchange data programmatically, as seen in Figure 6-49. Data transfers along the wires using the same broker as click-to-action. Besides saving the extra step of having the user click the data source to select a target, wiring portlets together enables greater flexibility to match brokered values.

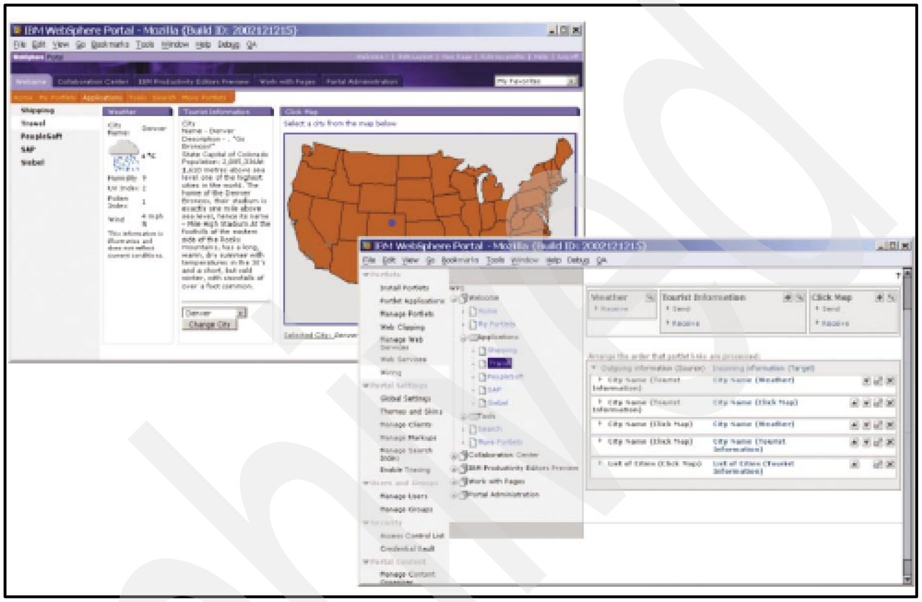


Figure 6-49 Pre-wired cooperation portlets

Discoverable services

The portlet API provides an interface to enable dynamic discovery of available services. Each service is registered in a portal configuration file and is accessed from the PortletContext.getService() method, which searches the factory for the service, creates the service, and returns it to the portlet. This method makes services available to all portlets without having to package the service code with the portlet. And you can exchange or enhance the implementation of this kind of service transparently without affecting the portlet.

The portal server provides discoverable services for its credential vault to manage persistent TCP/IP connections and to manage the portal content repository. Portal developers can implement new services as options, to further customize their infrastructure, such as location, notification, content access, or mail services.

6.3.3 Development tools

WebSphere Portal includes a range of development options, from non-programmatic business-user tools, Web clipping and Web services application-integration techniques, to Java 2 Platform Enterprise Edition (J2EE) technology-based portlet and portlet development tools to provide interactive access and data cooperation services across a range of Web, database, content management, enterprise resource planning (ERP), customer relationship management (CRM), and other solutions. More developer tools supporting portlet and portal development are available from IBM Business Partners.

IBM Rational Application Developer, Version 6.0 is a comprehensive integrated development environment (IDE), with full support for the J2EE programming model, including Web, Java, Web services, EJB, and portal application development. The product includes a set of visual portal development tools and a WebSphere Portal test environment, enabling you to build and test individual portlets and entire portal applications. Portlet wizards create a complete portlet that complies with the IBM Portlet API, as well as the new JSR 168 Portlet API, the industry-standard specification that addresses the requirements of aggregation, personalization, presentation, and security for portlets running in a portal environment.

The visual portlet-development tool enables you to build rich user interfaces for portlets quickly with Java Server Faces (JSF) components that generate code for event handling, user-input validation, and data handling. These tools also connect portlets to relational databases, Enterprise Java Beans (EJB) components, Web services, and enterprise information systems (EISs), such as SAP and Siebel, through point-and-click operations, as depicted in Figure 6-50 on page 207.

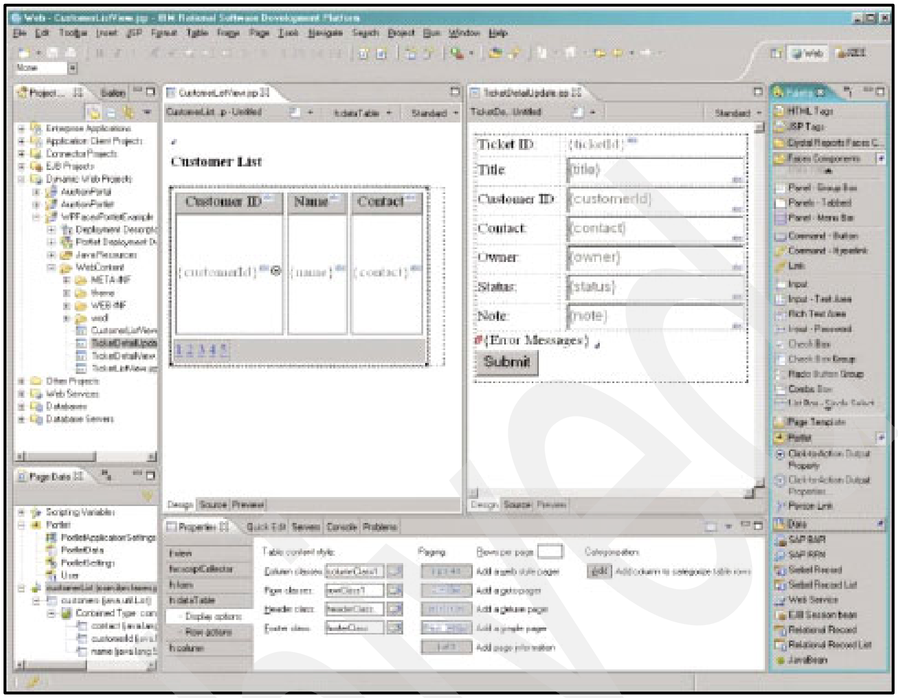


Figure 6-50 Visual Portlet development using Rational Application Developer

Rational Application Developer also provides the visual portal site development tools. The portal designer enables you to create portal pages and customize the layout of portlets and edit portal themes and *skins* that control portal site appearance. The created portal site can be tested on WebSphere Portal test environment or on a separate portal server as seen in Figure 6-51 on page 208.

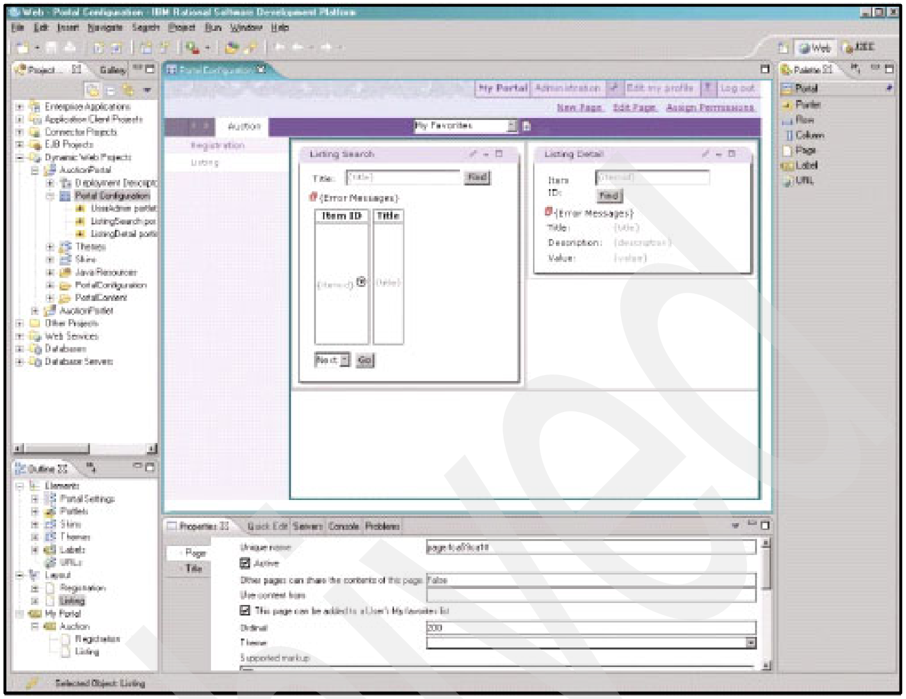


Figure 6-51 Portal designer feature in Rational Application Developer

6.3.4 Personalization

The WebSphere Portal personalization component provides features that allow subject matter experts to select content suited to the unique needs and interests of each site visitor. These Web-based tools help companies quickly and easily leverage content created by line-of-business (LOB) users and subject matter experts.

A personalization solution involves three basic components:

- A user profile, which provides information about users of the site, including user attributes
- A content model, which defines attributes about the content, such as product descriptions, articles, and other information
- Matching technology, which includes engines that match users to the right content and also provides filtering, rules, or recommendation engines or combinations of all three

The WebSphere Portal personalization engine and WebSphere Portal share a common user profile, a common content model, and the JSR 170 content repository. This model is based on the WebSphere resource framework interfaces classes. As a result, personalization rules can easily be added to portlets to select portal content from IBM Workplace Web Content Management and WebSphere Portal Document Manager and target it to the portal's registered users. A set of wizards is included with Rational Application Developer to access Structured Query Language (SQL) or LDAP data.

The basic steps associated with personalization involve classifying site visitors into segments and then targeting relevant content to each segment. Business experts create the rules for classifying users and selecting content using Web-based tools as shown in Figure 6-52.

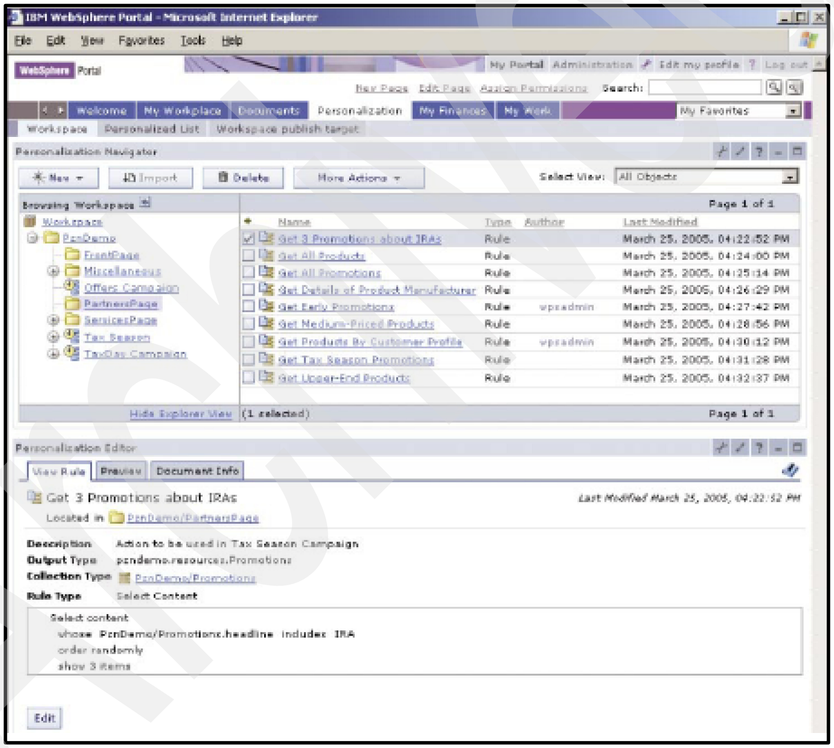


Figure 6-52 Creating business rules for personalization

The WebSphere Portal personalization recommendation engine provides collaborative filtering capabilities. *Collaborative filtering* uses statistical techniques to identify groups of users with similar interests or behaviors.

Inferences can be made about a particular user's interests, based on the interests of other members of the group.

The WebSphere Portal personalization engine also includes campaign-management tools. *Campaigns* are sets of business rules that work together to accomplish a business objective, as shown in Figure 6-53. For example, a human resources (HR) manager might want to run a campaign to encourage employees to enroll in a stock purchase plan. The HR manager would define a set of rules as shown to accomplish this business objective. Campaigns have start and stop dates and times and can be e-mail-based and Web-page-based. Several campaigns can run simultaneously and can be prioritized.

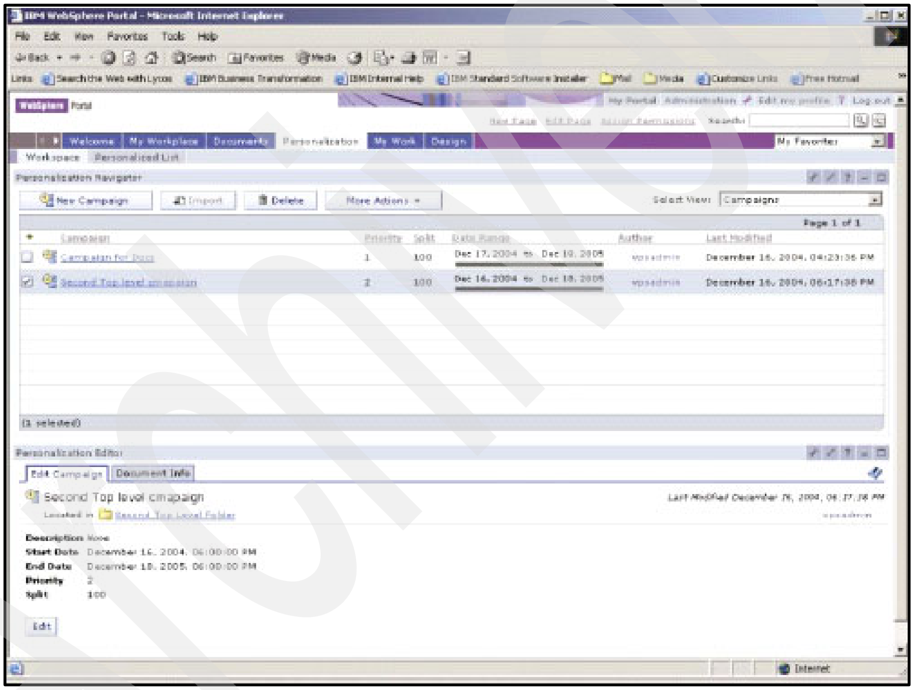


Figure 6-53 Campaign parameters

Implicit profiling services can collect real-time information about site visitor actions and then construct personalization business rules using this data. Implicit profiling tracks the areas of a site where a user is most active to determine the user's interests. For example, if a user clicks articles about football, it is possible to determine that he is interested in sports, even if he did not explicitly indicate this interest when registering at the portal. The personalization rules and campaigns are stored in the WebSphere Portal Java Content Repository (JCR) store, and you can write rules to personalize the WebSphere Portal Document

Manager or IBM Workplace Web Content Management content right out of the box (in other words, without developing your own resource collections).

To analyze the effectiveness of the site and its personalization strategies, the server provides a logging framework that can be enabled using WebSphere Portal administrative services. These logs can be analyzed by Tivoli Web Site Analyzer, included with IBM WebSphere Portal Extend. Tivoli Web Site Analyzer can then create reports for the portal business owner. This helps companies to measure the effectiveness of the business rules and campaigns in achieving their objectives. Personalization rules and campaigns, which are ready to use and enable you to write rules to personalize JCR content, are stored in the Java content repository.

6.3.5 Administrative portlets

Administrative portlets are provided to add portlets to the portal registry, manage users, groups, and access-control lists, integrate Web content with clipping technology, configure Web services-enabled applications, set portal-wide settings, and manage logs and other common tasks. The next section describes some of the administration portlets and what they do for you. Because some of these portlets have already been described in the previous section, this section focuses on those that have not yet been covered.

Global settings portlet

In the global settings portlet, administrators can change portlet settings such as the default language, the cache timeout values, and so on. Settings are also available that control how new user sessions are handled, and what to do when a user tries to access a portlet without authorization. Unauthorized access can be ignored (in other words, the portlet is not displayed), or the portlet can be replaced by an informative message so that the user can take the necessary actions to correct the situation. Returning users might want to pick up where they previously left off, so there is a setting to retain the state of the last visit and return to that page the next time.

Web clipping portlet

One of the most important portlets is the Web clipping portlet, which is used to display sections of existing Web pages. You can visually select portions of the page or clip all the text between specific tags. This method enables you to precisely control what markup is extracted and quickly enable access to external Web-site content sources from within WebSphere Portal. The portlet can optionally rewrite the links inside the clipped page, which is useful for displaying existing pages without leaving the portal's navigation structure. Each time you clip a Web page, a new portlet is created in the portal registry. Whenever the new portlet is displayed, it retrieves the current version of the Web page and extracts the clipped portion to display.

Web services

A *Web service* is an interface that describes a collection of network-accessible operations. WebSphere Portal provides support for the OASIS WSRP standard. The OASIS WSRP standard simplifies the integration of remote applications and content into portals. With WSRP, you can select from a rich choice of remote content and applications (presentation interfaces and their data) and integrate them into your portal with just a few mouse clicks and no programming effort. You can also configure WSRP producers support for specified WSRP-enabled local portlet applications as presentation-oriented WSRP services and make them available to remote portal server consumers.

The WSRP interface is described using a standard XML description language called Web Service Description Language (WSDL), so that the service can be invoked without prior knowledge of the platform, language, or implementation design of the Web service. Web services are located using the standard Universal Description and Discovery Interface (UDDI), which can be a private or public registry of Web services.

The Web services administration portlet enables administrators to configure Web services producer and consumer support for specified WSRP-enabled applications within the portal.

Support for WSRP also enables WebSphere Portal to integrate external applications through remote portal servers and to promote local portal portlet applications to remote portals. A new portlet can be programmatically activated, but with no special permissions. Access control for the new portlet is inherited from the default portlet settings but can be set explicitly by the administrator.

User and group portlets

You can use the portlets provided to manage user and group information without leaving the portal. You can also manage the group memberships of a user. These portlets provide search capabilities and pagination that enable the administrator interface to scale and manage a large number of users and groups.

The portal server uses group-membership information to determine which pages, portlets, and documents a user is authorized to view and edit. Users can be members of one or more groups, and groups can contain other groups. Users are allowed access to portal resources when access is granted to any group to which the user belongs. Access rights can also be granted to specific individuals, but most companies find it easier to manage the access rights of groups.

Portal analysis and logging portlets

You can control the tracing and logging task and monitor user activity through the portal analysis administrative options and portlets that enable tracing and track frequent-user information. You can also control tracing and logging activity by modifying the configuration properties files of the logging subsystem. The portal server also records user activity in logs that can be processed by Tivoli Web Site Analyzer. Overall usage statistics such as logins and logouts are tracked, along with portlet and page-usage statistics.

IBM Common PIM portlets

WebSphere Portal, Version 5.1 introduces IBM Common PIM (Personal Information Management) portlets, which are designed to provide a common portlet interface to interactively work across a number of popular mail and PIM applications, including Lotus Notes and Domino, Microsoft Exchange, IMAP, and POP3 mail applications. Common PIM portlets integrate with WebSphere Portal applications and services, including the WebSphere Portal rich-text editor, to create and edit mail messages and documents. The portlets include the ability to view received attachments using the portal attachment viewers and the ability to store and retrieve documents from the WebSphere Portal Document Manager. Presence-awareness capabilities are also available to provide access to instant messaging and chat services.

Application-integration portlets

By definition, a portal provides access to content, data, and services located throughout the enterprise. These include not only predefined connectors and portlets but also other connectors and portlets that can be created by various tools. ERP and CRM systems are excellent candidates for these portlets because efficient, personalized access to these functions provides measurable return on your portal investment.

WebSphere Portal includes portlets and portlet builders that help you access a variety of ERP and CRM systems. IBM WebSphere Portal Application Integrator is a non-programmatic portlet-building tool, provided with WebSphere Portal, that offers a fast way for power business users to build new portlets that access various kinds of enterprise application systems, including portlet builders for SAP, Siebel, Oracle, and Domino applications, and Java Database Connectivity (JDBC) sources. It also works for relational databases, such as DB2, Oracle, and others. WebSphere Portal Application Integrator portlet builders work by querying the back-end system for metadata that describes the business object of the specific enterprise application that the new portlet is meant to work against.

WebSphere Portal Application Integrator portlet builders support the WebSphere Portal credential vault service to store and retrieve user credentials and pass them on to the enterprise application and business object to manage single sign-on for the portlet user with the respective enterprise application. After the selections are made, the portal server stores the configuration information needed to run the portlet. No new Java code or service needs to be deployed, just the configuration information for each portlet.

Using this approach, anyone who understands the usage pattern of the enterprise application system can build a portlet in a short time and optionally include presence awareness and data exchange between portlets and enterprise applications using click-to-action.

Collaboration portlets

Corporate portals connect people to the applications, content, and resources they need. Portals also connect people to each other through community pages, shared bookmarks, contact lists, and personal productivity portlets. Collaboration is really about people working efficiently and effectively in teams to solve business problems. WebSphere Portal includes out-of-the-box, preconfigured pages, portlets, and services designed to support the team and its activities, with messaging, document libraries, user profiles, in-boxes, calendars, online directory services, team workplaces, and electronic meetings. Users can access these collaborative services in the context of what they are currently doing, instead of having to leave the portal to open another application.

A portal user can see directly from a portlet if other users are online and select from a menu of options to interact with another user. For example, while working in the portal, users can easily see who else is online and then send an instant message, send an e-mail, or add a person to their contact lists. Collaborative portlets have advanced built-in features that allow portal users to take actions on documents or user names that appear in a portlet.

6.4 WebSphere Business Modeler

WebSphere Business Modeler products help organizations fully visualize, comprehend, and document their business processes. They are based on Eclipse technology. Rapid results can be obtained through the collaboration functionality, where subject matter experts team to define business models and eliminate inefficiencies. You can model business processes, deploy, monitor, and take actions based upon key performance indicators (KPIs), alerts, and triggers for continuous optimization. Business processes then get aligned with strategic corporate objectives and honed as required. WebSphere Business Modeler products can drive much more granular business insight and knowledge, where knowledge equates to competitive advantage.

WebSphere Business Modeler products can serve to close the gap that exists between organization business units and IT understanding of the business drivers. IT requirements for defining a system are well articulated, since both are utilizing a common framework implemented through an Eclipse 3.0 shared workspace. Given that a business process is a defined set of activities leading to specific results, modeling them assures that your best practices are well documented and communicated. Accurate business modeling is the starting point for successful IT deployment.

The business measures editing function allows you to define the KPIs and metrics for your organization. When modeling your information assets, the information model provides a view of your data and its exact use within a business process. The resource model allows you to identify all of the different resource types so they can be associated directly to the model. The swimlane view can be used to visually display the model delineated by role, resource, organization unit, and classifier. You can both view and edit models in the swimlane view.

The WebSphere Business Modeler solution delivers a structure for continuous business improvement. A core underlying functionality is the ability to analyze and model business processes. WebSphere Business Modeler provides software tools to help you model, simulate, and analyze complex business processes. Line-of-business managers as well as business analysts can design and develop process flows that improve how you do business. The WebSphere Business Modeler products can help you maximize business effectiveness by optimizing processes that give you competitive leadership in today's On Demand Business environment.

The four products in the V6 WebSphere Business Modeler family are:

- WebSphere Business Modeler Basic: Provides a low-cost option for business users who are looking for a simple and easy-to-use tool to model, document, and print their business processes.
- WebSphere Business Modeler Advanced: Provides all the features of WebSphere Business Modeler Basic, plus complex model simulation and analysis capabilities. In addition, it lets IT-focused users export models to multiple buildtime environments to help jump-start application development.
- WebSphere Business Modeler Publishing Server: Provides the ability to publish business process models to a Portlet-based server where multiple subject matter experts can view and review the information simultaneously through a standard Internet browser.
- WebSphere Business Modeler Publishing Edition: Consists of 10 WebSphere Business Modeler Advanced licenses, plus one license of the WebSphere Business Modeler Publishing Server.

6.4.1 Advanced

WebSphere Business Modeler Advanced provides functionality for process modeling, enterprise modeling, essential data and artifact modeling, organization modeling, resource modeling, time line and location modeling, and business process analysis. The WebSphere Business Modeler can be used in five modes:

- Basic
- Intermediate
- Advanced
- Business Process Execution Language (BPEL) for either the WebSphere Business Integration Server Foundation or WebSphere Process Server runtime
- WebSphere MQ Workflow Flow Definition Language (FDL)

With WebSphere Business Modeler Advanced, business and IT communities utilize an Eclipse-based shared workspace. BPEL and FDL allow you to work in the mode that you will deploy. Business analysis is core for improving businesses. Many business process modeling efforts stop at developing flow diagrams or process maps. With WebSphere Business Modeler Advanced, the effort is extended to include simulation, analysis, and redesign.

WebSphere Business Modeler Advanced has a visual process modeling environment that is enhanced by the ability to color code the elements by role, classification, or organization units. In addition to the name of the object, labels can be added to the top and bottom of an object. User customization allows the appropriate labels to be displayed on the model at the appropriate time. In addition to these visual analysis techniques, a swimlane view can be used to display the model delineated by role, resource, organization unit, and classifier. WebSphere Business Modeler V6 provides the ability to view the model via the swimlane and also edit the model in the swimlane view.

WebSphere Business Modeler Advanced includes a simulation engine that allows you to simulate the behavior of your process, permitting analysis of workloads and bottlenecks. You can view analysis on the process, resources, activity, and queue during the simulation or after the simulation is complete. The flows are animated in a step-by-step simulation that allows you to see simulated data on the flows. Simulation snapshots are available for reference. The simulation engine enables you to determine the most efficient model prior to implementation by quickly conducting the *what-if* analysis. To quickly go from the modeling phase to the simulation phase, metric information (such as costs and time) are entered into the model while it is being modeled in the process editor.

WebSphere Business Modeler Advanced provides capabilities for adding and viewing simulation attributes in a table format to simplify the steps to add the attributes and run a simulation. In addition, new distributions have been added to support additional statistical types of analysis that are applicable across industries. Another important feature to support the simulation capabilities is the ability to generate multiple instances of a single resource to faster simulate resource utilization within a business process.

WebSphere Business Modeler Advanced includes a reporting tool that can programmatically create written, numerical, and graphical reports. The reports provide information used in the process analysis and redesign. Some of the predefined reports are:

- Process Summary Report: Provides a single report that contains essential cost and time analysis reports.
- Process Comparison Reports: Combines and compares the Process Summary Reports from two process simulations for comparisons and provides ROI comparisons of as-is and to-be flows.
- Documentation Report: Provides an overview of the attributes of business items, resources, or other model elements.
- Procedure Report: Documents the sequence of steps within a process and the relationships of that process to other processes and roles.

All of this can be rolled up into communication tools through charts, tables, reports, graphs, or diagrams. If a customized report is needed, there is a Report Builder, which also supports publishing models to the Web, to help you generate it.

WebSphere Business Modeler V6 includes integration with Crystal reports, which enables you to create reports that can combine business process information with additional business information.

Once the models have been built, simulated, analyzed, and optimized, you can export them to multiple code generation tools. The models can be exported to WebSphere Studio Application Developer Integration Edition V5.1 through BPEL, WSDL, and XML schema definitions. In addition, WebSphere Business Modeler exports to the new WebSphere Integration Developer tooling environment through the generation of Service Component Architecture (SCA) components, modules, task execution language (TEL) for human tasks, BPEL, WSDL, and XSD. WebSphere Business Modeler Advanced can also create the wiring of those artifacts. WebSphere Business Modeler Advanced also supports the capability to export to Rational XDE under the Business Modeling Profile. Integration with Rational Software Architect (RSA) is performed within the RSA environment by importing the WebSphere Business Modeler process models into the RSA tool. And finally, the tool still supports integration with WebSphere

MQ Workflow through flow definition language (FDL). FDL can be exported directly to WebSphere MQ Workflow Buildtime.

In addition to the deployment capabilities, WebSphere Business Modeler provides model management capabilities by being able to version, check in and check out, merge, and compare different versions, track element history, and get a specific version from the history. These capabilities are provided by the CVS plug-in as part of the Eclipse platform. WebSphere Business Modeler offers optional support for Rational ClearCase as a version control system.

Another important feature of WebSphere Business Modeler is the ability to generate business measures models via the business measures editor for export to Business Monitor. This feature enables WebSphere Business Modeler to support WebSphere Business Monitor by allowing you to create a business measures model of a modeled process and update it with key performance indicator (KPI) and metric information. KPIs and metrics are modeled by defining what triggers them and under which conditions they are monitored. In addition, situations can be defined that allow WebSphere Business Monitor to determine a situation has occurred and an action must be taken. Once you have defined the KPIs, metrics, and situation, an integration specialist can ensure that the appropriate events both are available and contain the correct information in order to properly be monitored. Once this has been completed, a business measures model is generated. This business measures model is a deployable artifact for the WebSphere Business Monitor.

WebSphere Business Modeler Advanced is able to import FDL from WebSphere MQ Workflow, models from WebSphere Business Modeler V4.2.4, and WebSphere Business Modeler V5 models. Included with the product are Visio import capabilities as well as a proprietary XML import and export capability. This XML capability can be used to facilitate integration with other tools, which allows users to import information from external sources as well as export it to additional environments.

WebSphere Business Modeler allows you to support and implement enterprise initiatives, such as:

- Workflow automation
- Policy and procedure documentation
- Application development
- Sarbanes-Oxley initiatives
- HIPAA

WebSphere Business Modeler Advanced provides support for:

- WebSphere MQ Workflow by exporting in FDL
- WebSphere Server Foundation by exporting in BPEL
- WebSphere Process Server by exporting BPEL
- Rational XDE through UML 1.4 export

6.4.2 Basic

WebSphere Business Modeler Basic is a low-cost entry edition providing basic modeling support. The Basic Edition is for enterprise departments and individual users who do not require the full capabilities of WebSphere Business Modeler Advanced. WebSphere Business Modeler Basic is used for modeling, validation, and documentation. All three modeling modes (basic, intermediate, and advanced) are available. Once the models are entered, the validation rules still apply; however, WebSphere Business Modeler Advanced is needed to utilize the simulation engine and analysis. The reporting tool and complete set of static documentation reports are available in this base release. The models can be captured, validated, and documented in this version. WebSphere Business Modeler Advanced is needed to run simulations and export to the multiple code generation tools.

Table 6-5 on page 220 describes the key functional differences between WebSphere Business Modeler Basic and WebSphere Business Modeler Advanced.

Table 6-5 Basic and Advanced comparison

Feature	Basic	Advanced
Modes	Basic, Intermediate, and Advanced	Basic, Intermediate, Advanced, BPEL, and FDL
Versioning	Yes	Yes
Simulation	No	Yes
Static and Dynamic Analysis	No	Yes
Reporting (create reports)	Yes	Yes
Query Report	Yes	Yes
Basis Report Templates	Yes	Yes
Printing	Yes	Yes
Modeler Project Import/Export	Yes	Yes
Delimited file Import/Export	Yes	Yes
ADF Import	Yes	Yes
XSD Import/Export	No	Yes
UML Export	No	Yes
FDL and BPEL Export	No	Yes
FDL Import	No	Yes
Visio Import	Yes	Yes
XML Import/Export	No	Yes
Swimlane Editor	Yes	Yes
Business Measures Modeling	No	Yes

6.4.3 Publishing Server

WebSphere Business Modeler provides the capability to publish business processes created within WebSphere Business Modeler. Once published, authorized viewers can view and comment on the business processes using a standard Internet browser.

When publishing the business process, WebSphere Business Modeler is able to create visual representations of business processes along with supporting information. The more accurate and detailed the information contained in WebSphere Business Modeler, the more accurate the analysis. An important step in modeling the business processes is to validate the process and its data. One way to achieve this validation is to make the business processes available so that subject matter experts and other interested parties can review them. Publishing Server allows someone using WebSphere Business Modeler to publish an entire business process modeling project or just parts of it to a server. Models are published in the standard process model view. The subject matter experts and other reviewers can then view the process diagrams and their supporting information using a standard Internet browser.

In addition to just viewing business processes, authorized reviewers can use Publishing Server to comment about a diagram or about its supporting information. Because the comments are visible and other reviewers can respond to the comments, Publishing Server provides a forum for discussing and resolving differences of opinion on a business process. The comments and responses are tracked by originator and date/time at creation. These comments and responses can then be exported permitting the modeling team to update the model, making it more accurate.

WebSphere Business Modeler Publishing Server also allows authorized reviewers to post attachments such as Microsoft Word documents. The attachments allow the reviewer to provide the modeling team with additional information that they can use to update the business processes.

WebSphere Business Modeler Publishing Server has three major components: publisher, publishing server, and client.

Publisher

The publisher is a plug-in for Business Modeler that takes modeling projects or selected elements from a modeling project and publishes them to the publishing server.

The publisher acts as a gateway to the publishing server. The person performing the role of publisher selects which projects or model elements to publish and which publishing server to host the published project. Projects or model elements can be published in an in-progress mode. An administrator can then publish the information into a released mode. Information published in both modes is displayed with information contained in the advanced mode of WebSphere Business Modeler. This becomes useful when working with multiple versions of the business process models so that the correct level of information is shown to users at the right time.

Publishing server

The publishing server hosts published business modeling projects. The server consists of WebSphere Portal Server and a database. WebSphere Portal Server displays the business process information in four portal windows and an additional portal for displaying comments. The database stores the comments.

The publishing server also has an administration page that allows an authorized person to perform the administration role to define user access, edit and remove business processes, and edit and remove comments.

Client

The client is a standard Internet Explorer Internet browser with the Scalable Vector Graphics (SVG) Viewer plug-in. The SVG Viewer enables you to zoom in and out of the process diagrams.

To access a published business process model, the person performing the role of reviewer enters the model's URL on the publishing server. Once the reviewer has logged onto the publishing server, the client displays the portals served by the server. The information displayed by the portals depends on what elements or buttons the reviewer has clicked in any of the portals.

When viewing published business process models, once a process or element has been selected, you can select information either through the navigation portlet or the view portlet. Once a business process model or model element is selected, you can view its associated attribute information as well as any comments or responses or add new comments and responses via the comments portlet. In addition, you can associate additional read only documents or URLs to model elements in the view portlets to help provide additional contextual information about the element.

6.5 WebSphere Business Monitor

The WebSphere Business Monitor Version 6.0 is a Web-based client/server application that measures business performance, monitors processes and workflow, and reports on business operations. It monitors business processes at runtime by monitoring event-emitting runtime engines. The information captured can help you identify problems, correct faults, and change processes to achieve a more efficient business.

WebSphere Business Monitor calculates key performance indicators (KPIs) and metrics using collected events, based on a given model. The calculated KPIs and metrics values are represented on a number of views based on business needs. Users are notified of incidents requiring their attention and corrective actions can be performed to avoid failures. It supports different notification methods (such as alert, e-mail, cell phone, pager, and service invocation) against situations and actions associated with defined conditions.

WebSphere Business Monitor includes one copy of WebSphere Business Modeler Advanced providing modeling and simulation capabilities to model the critical aspects of businesses. WebSphere Business Monitor is used in conjunction with WebSphere Business Modeler to create business measures models that specifically identify activities to be monitored, including events, business metrics, and KPIs.

Features of WebSphere Business Monitor 6:

- Captures a large amount of data through events from operation activities and transforms it into metric and KPI values
- Extracts the measurements variables from business data
- Displays the measurements values in useful views
- Provides analysis and reports
- Performs corrective actions
- Notifies users to take action to prevent failures
- Customizable dashboard, implemented as WebSphere Portal pages, that are visually intuitive (scorecards, key performance indicators (KPIs), and gauges)
- Multidimensional analysis and reports supported through the dashboards, with embedded business intelligence
- Customized analytic components that monitor existing business processes
- Allows user to do multidimensional analysis
- Capability to monitor specified by the business user
- Adaptive Action Manager invokes selected real-time action or set of actions based upon predefined rules and policies
- User-controlled filtering of the reports
- Sophisticated business analysis integrated with business processes
- Innovation and optimization enhanced in key business processes
- Capability to understand and transform your business through insightful business modeling, simulation, and analysis enabled with WebSphere Business Modeler Advanced

Business measures models are used for monitoring procedures. These models are created in the Business Measures editor where you can specify the measuring points and event filters, define the measurements, their correlations, and sources of the business data. When the business measures model is complete, you can export it to WebSphere Business Monitor. It then recognizes the model to be monitored and the measurements to be captured from the incoming events.

The business measures editor is used to open the process models created in WebSphere Business Modeler and to create business measures models. For each business measures model, you can define the metrics and KPIs, event emission points, event filters, event composition rules, and situations that trigger specific actions at runtime.

Once the business measures model is complete, work can be performed to enable it to be recognized by WebSphere Business Monitor. Then, the Business Monitor understands the measurements to be captured from incoming events.

In addition, continuous business process improvement metrics, such as task working durations and process decision percentages are calculated and can be exported to update their corresponding business process models in the WebSphere Business Modeler. This improves simulation and analysis results because now the actual metrics (versus assumptions) are factored into the model. These capabilities provide for processes running in WebSphere Process Server, which is described in more detail in 6.6, “WebSphere Process Server and Integration Developer” on page 235.

Many business process modeling efforts stop at developing flow diagrams or process maps. With WebSphere Business Modeler Advanced, this effort is extended to include simulation, analysis, and redesign.

6.5.1 Architecture

The architecture of WebSphere Business Monitor 6.0 comprises a set of internal components and a group of external components.

The diagram shown in Figure 6-54 on page 225 represents the overall logical reference architecture and the components of Monitor V6.

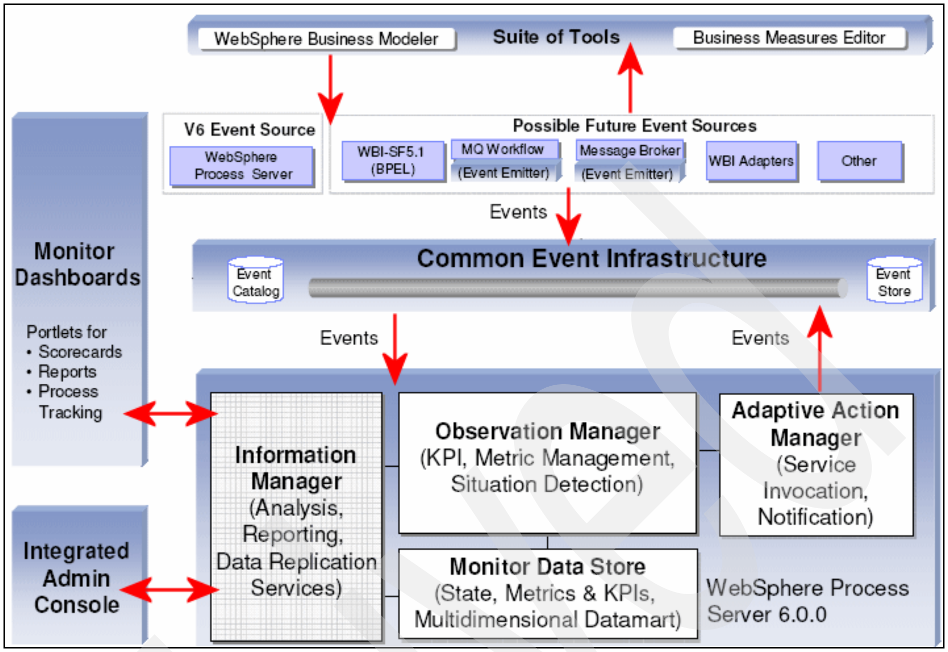


Figure 6-54 WebSphere Business Monitor architecture

Internal components overview

The following is a list of the internal components:

- **Monitor server:** Receives events, handles monitoring-context instances, and stores and persists runtime and historical metrics and KPI values of those instances.
- **Dashboards:** Display the monitored data. They provide a predefined set of views that can be customized to support different representations of data and offer enhanced data analysis.
- **Schema Generator:** Generates database scripts to be used for creating databases tables in state, runtime, and historical databases. These databases contain the business measures models data. The schema generator also generates the DWE OLAP™ metadata description of the historical database and generates the metadata mappings for the replication manager.
- **Databases:** Provide the Monitor server with information for event processing. They also provide the dashboard client with information for populating views. Information is transferred across the databases through another monitor component, the replication manager.

- **Adaptive action manager:** Provides different types of business responses resulting from situations expressed within the incoming events.

External components overview

The following is a list of the external components:

- **Business measures editor** (BME): It is used to create the business measures model that defines what should be monitored, for example, monitoring contexts, key performance indicators, metrics, and business situations.
- **Common event infrastructure** (CEI): WebSphere Business Monitor uses the Common Event Infrastructure (CEI) and the Common Business Event (CBE) format. This means that the WebSphere Business Monitor leverages the CBE format for consuming and emitting events.

  It participates in event management by receiving events from event sources and transferring them to the event consumers that have expressed interest in those events.
- **DB2 Alphablox and DWE OLAP:** Provide enhanced data analysis for dashboards.

Figure 6-55 on page 227 is an overview of the components.

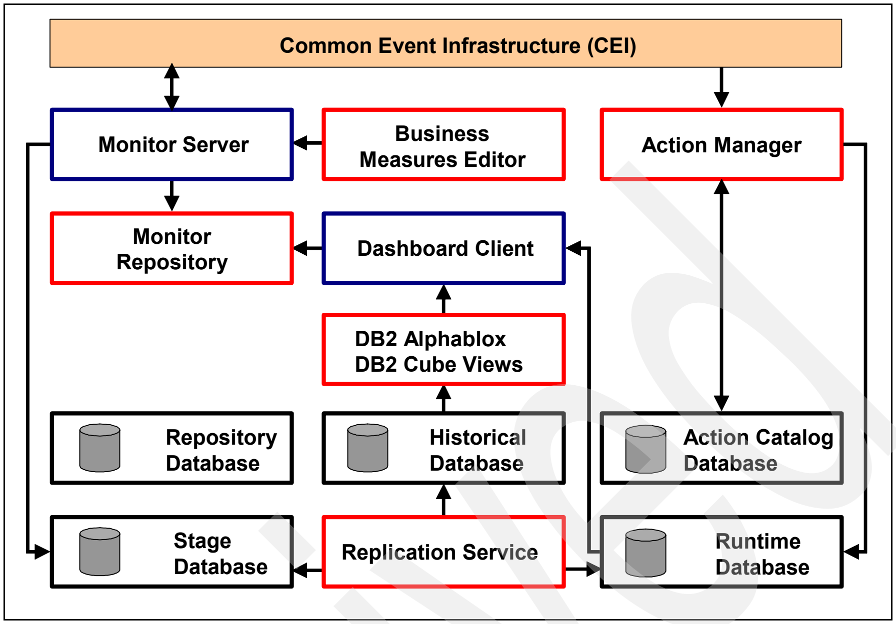


Figure 6-55 WebSphere Business Monitor component overview

6.5.2 Component details

WebSphere Business Monitor 6.0 enables you to monitor the runtime behavior of business processes through a Web application that is deployed on WebSphere Process Server V6.0. The data that it monitors is issued from a runtime engine.

Business measures model

Monitoring data is based on a business measures model, which includes artifacts that permit correlation of the runtime events with a specific instance, in addition to entries that specify situations. The business measures model is obtained from the original business model by editing entries that are essential for monitoring purposes: correlation of events, metric calculations, and detecting situations. Editing of the model is done by the Business Measures Editor.

Event handling

Data is encapsulated in common base events by means of event emitters, which are transmitted on a common event bus, the common event infrastructure (CEI). For WebSphere Business Monitor V6, only events emitted from WebSphere Process Server are supported.

WebSphere Business Monitor V6 runs on WebSphere Process Server V6, which sits on top of WebSphere Application Server V6. The runtime engine (WebSphere Process Server) for the real application runs on another machine and CBEs are extracted from the engine and submitted on the event bus, which is configured as a service integration bus (SIB) between the two machines. The Monitor Server receives the common base events from the SIB, matches them to monitoring context instances, and calculates the appropriate metrics and KPIs to ultimately display for the user on the dashboard.

The common event infrastructure (CEI) is a shared component that can operate either inside or outside of WebSphere Application Server. The CEI provides facilities for the runtime environment to persistently store and retrieve events from many programming environments. Events are represented using the common base event model, a standard, XML-based format that defines the structure of an event. The events are passed through JMS across the service integration bus (SIB).

WebSphere Business Monitor server

WebSphere Business Monitor server is used to perform the management of the KPIs and metrics, as well as other relevant information from the business measures model, which is defined as part of the business measures modeling capabilities of the WebSphere Business Modeler. The monitor server is also responsible for event correlation as well as situation detection that then triggers the adaptive action manager component.

Adaptive action manager

The adaptive action manager (or action manager) is another key component in the WebSphere Business Monitor 6 product architecture, which provides the *sense and respond* functionality. The adaptive action manager is the WebSphere Business Monitor component that receives situation events emitted by the observation manager. It selects appropriate actions based on predefined bindings between the situations and actions that are set by the user and invokes one or more action services. The CEI sends these situation events to the adaptive action manager, which parses them, selects appropriate actions based on predefined rules created by the user, and invokes a selected action or set of actions.

The adaptive action manager performs two types of actions: notification actions and service invocation actions:

- Notification actions take the form of e-mail, SMS, pager message, or a dashboard alert.
- Service invocation actions invoke a Web service, or a BPEL process through a Web service invocation.

The adaptive action manager parses the received situation event and selects an appropriate action by looking up the action in the action catalog database, where action-related information and binding information are stored. If the appropriate action is a dashboard alert, the action manager extracts the data needed for the creation of the alert-notification record from the received situation event and inserts this record in the WebSphere Business Monitor Runtime database. The record is collected by the Alerts view in a dashboard. The adaptive action manager uses LDAP as the user-registry for user notifications.

6.5.3 Databases

The data architecture of WebSphere Business Monitor V6 has been optimized for both transaction processing data stores as well as data marts for reporting, analytics, and business intelligence.

Simply stated, the WebSphere Business Monitor V6 is responsible for its own data store to handle data required for the monitoring operation: instances of running monitoring contexts and metric values. The performance is optimized by dividing the data store into different databases, and each database is optimized for specific types of database (DB) access operations.

DB2 replication services are responsible for moving state data to the historical data store at configurable replication intervals. This fundamentally separates the transaction processing data store from the historical data store for high-performance event processing.

Data analysis can then be performed on the historical data, made available by introducing DWE OLAP and accessing cubes from DB2 Alphablox interface, which is the visualization module.

The database topology for the Monitor Server and Dashboard Server in a given server environment can vary, for example:

- The Monitor Server runs on its own machine with the State and Repository databases.
- The Monitor Dashboard runs on a separate machine using WebSphere Application Server and Portal Server with the Runtime and History databases.

This setup is done for performance reasons. However, you might want the Repository database on the Dashboard Server because the Monitor Server only uses the Repository database at the time that you import a model. However, the Dashboard Server uses it frequently when configuring and displaying the dashboards.

The WebSphere Business Monitor uses a number of databases to store event information. Here is a short description of the databases.

Repository database

The Repository database contains the metadata describing the currently deployed business measures models as well as information about the other WebSphere Business Monitor databases. The Repository database contains the history of the deployed models. There is only one Repository database per WebSphere Business Monitor installation.

The Repository database is used by the Launchpad, which populates it with the database attributes for the State, Runtime, and Historical databases. These attributes are the database name, database schema, and host names of the database server. They are used by the other WebSphere Business Monitor components to access the State, Runtime, and Historical databases at runtime. The Repository database is also populated during the import of the business measures model.

State database

The State database stores information about running instances. This information includes metrics, business measures, and key performance indicators (KPIs) values. It is optimized for heavy transaction workloads. There is only one State database per WebSphere Business Monitor installation.

Each process instance requires two tables in the State database to store metrics, business measures, and KPIs. The structure of these tables is as dynamic as the structure of the process instance. Each business measure is represented by a separate column in one of the two tables. Depending on the options selected during the building of the business measures models, much or all of the information in the State database is replicated to the Runtime database.

The State database is used by WebSphere Business Monitor server. At runtime, the WebSphere Business Monitor server inserts, retrieves, and updates the information of processes instances that reside in the State database, according to the processed events.

The State database stores the following information:

- Information about business measures groups, which is a part of the data in the imported business measures models.
- The running process instances that are created while the WebSphere Business Monitor is running.
- The event entries of the running processes. The *event entry* is the event data that is received for updating a specific business measures group.

Runtime database

The Runtime database is similar in structure to the State database. It receives replicated information from the State database about the current status of all running processes as well as the final status of recently completed or failed processes. This information is used by WebSphere Business Monitor dashboards. The Runtime database is also used by the Adaptive Action Manager to store alert notifications. There is only one Runtime database per WebSphere Business Monitor installation.

The information in the Runtime database is replicated from the State database. The Runtime database stores:

- Alert notifications sent by the Adaptive Action Manager to the dashboards
- Process instance data
- Metric values

The Runtime database is used by the WebSphere Business Monitor dashboards. The dashboards retrieve the running or recently completed instances data required to populate the views from the Runtime database. The dashboard views use the Runtime database for analytical purposes, so it is optimized for query processing and aggregate query processing.

All completed instances remain in the Runtime database for 24 hours and are deleted afterwards. Twenty-four hours is the default retention policy, which you can modify as part of the data movement service configuration.

History database

The History database stores all completed and running process instances. It is used by the dashboards for enhanced data analysis using DB2 Alphablox. There is only one History database per WebSphere Business Monitor installation. The data in the History database is never deleted.

The History database should only contain two years worth of historical data. This is one of WebSphere Business Monitor product requirements. As mentioned before, the historical data is never deleted automatically, so the DBA is responsible for deleting the data that is more than two years old.

The History database stores the information regarding long-running instances as well as completed instances. This information is stored as star schemas rather than in the flat transactional forms used in the State and Runtime databases. The History database is optimized for aggregated and long running queries. It is used by DB2 Alphablox in dashboard views to provide advanced multidimensional reports.

The information in the History database is replicated from the Runtime database.

The History database contains dynamic tables that are created according to the deployed business measures model. The schema generator generates the

History database schema, which is used to create dynamic tables and DWE OLAP definitions.

The History database is used by the WebSphere Business Monitor dashboards. The dashboards retrieve the data required to populate some views from the History database. For example, the Reports view focuses on analyzing data extracted from the History database.

6.5.4 The dashboards

WebSphere Business Monitor has WebSphere Portal-based dashboards that can be customized with a predefined set of configurable views that are implemented via portlets. A dashboard designer with WebSphere Portal skills can take these portlets and configure them or create custom dashboards. Some of these view types leverage DB2 Alphablox for a more refined data analysis.

Measuring the process using the Dashboard Client component of WebSphere Business Monitor Version 6.0 enables users to monitor business performance through a set of views. The following are examples of those views, which can be combined to create dashboards.

Scorecard view

A scorecard is a set of performance measures that are linked to objectives and goals of a specific business area. Business users select the KPIs pertaining to their various areas of responsibility and place them in *perspectives* (categories) on their scorecards. On the Scorecards view, users can easily watch the runtime values of vital numeric KPIs, monitoring them against their defined targets. This is depicted in Figure 6-56.

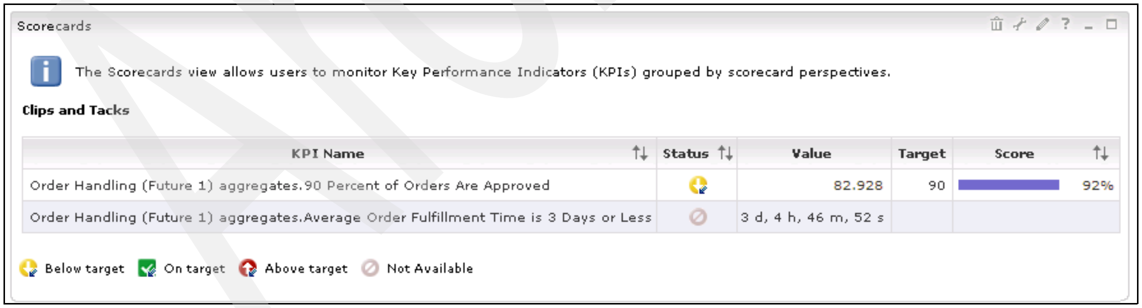


Figure 6-56 Scorecard view

KPI view

This view shows the values of individual KPIs, focusing on the position of the KPI value relative to its defined limits. It displays all KPI information so that business users can easily monitor the KPIs and take action if necessary, as depicted in Figure 6-57. For this, you must have a business measures model that contains the KPIs to monitor with their threshold limits.

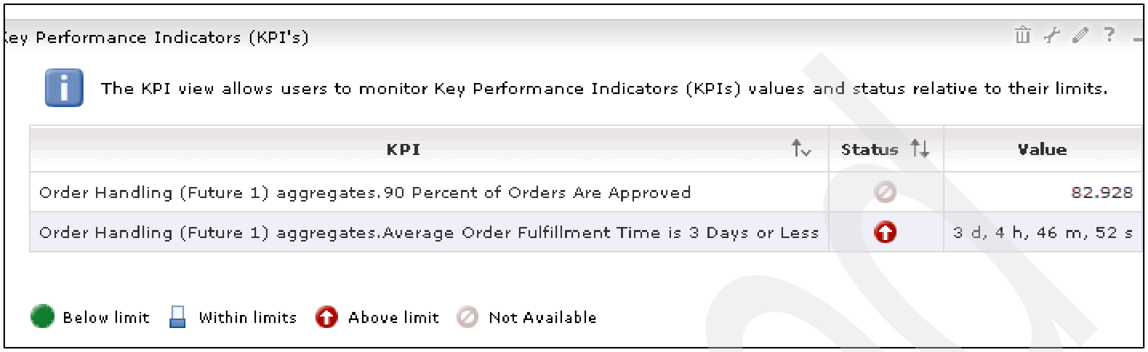


Figure 6-57 KPI view

Gauge view

This view displays individual KPI values, either relative to KPI threshold value or relative to the KPI target value. It visually uses a paradigm of physical instruments in the form of gauges and dials, such as automobile speedometers or temperature gauges. This is depicted in Figure 6-58.

Gauges help you to visualize information by representing KPI values. They have a focus on representing numeric KPIs that belong to aggregated business measures in a business measures model. Each gauge represents the value of a single KPI.

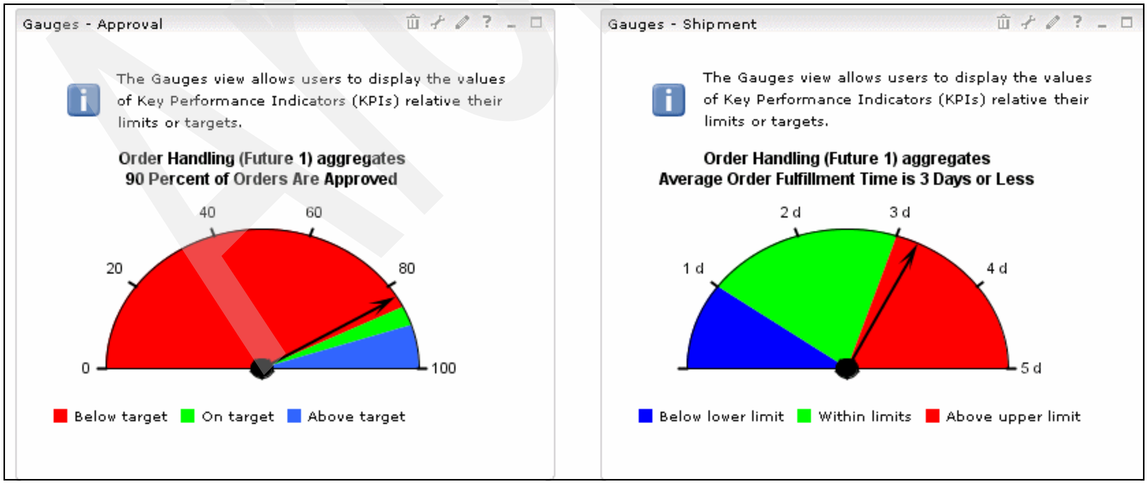


Figure 6-58 Gauge view

Active instances view

This view shows the details of processes, which contain a group of related metrics, KPIs, and situations in a particular process. The monitor view can monitor either individual instances or groups of process instances. This is in addition to providing the capabilities to perform drill-down and drill-up among child and parent instances.

The Active instances view shows the details of a process at runtime and displays information about running instances. You can monitor values of KPIs and metrics that belong to an aggregate business measures group, along with stopwatches and counters. You can also drill down to view the underlying activities, such as items in process instances, and whether they are realized by activities, local subprocesses, or global subprocesses.

The Active Instances portlet is shown in Figure 6-59, and consists of an Active Instances table that displays the business measures of each currently running process instance.

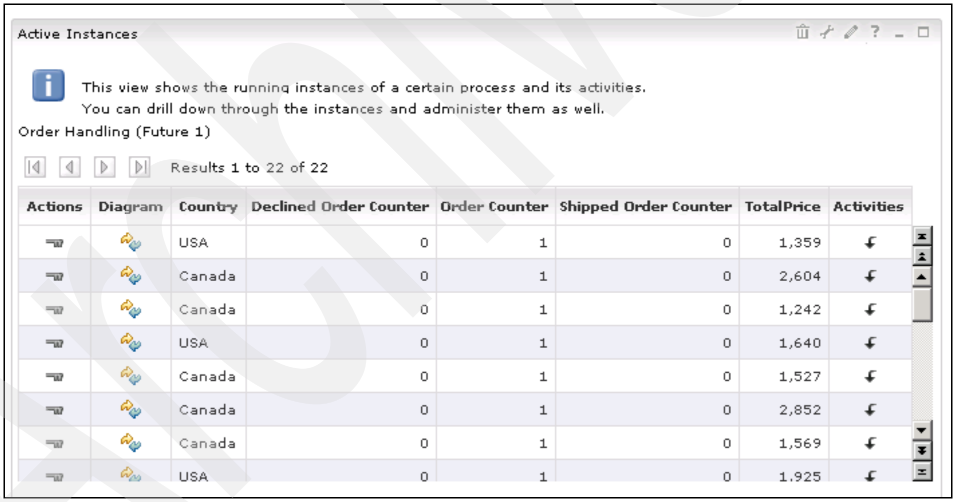


Figure 6-59 Active instances view

Other views are:

- Alert view displays the notifications for a specific user.
- Report view provides performance reports relative to a time-axis. Such reports contain tables and graphs for analyzing historical data contained within the performance warehouse data store. The Report view has built-in analysis types that include Quartile, Trend, and Control Analysis.

- Dimensional view provides a multidimensional view of business performance data. Users can pivot on business dimensions to view the performance.
- Process view displays the process status. This type of display shows a process graph with visual cues on the status of the process.

In addition to the previous views, there are some helper views such as:

- Organization view: Displays the organization, organizational units, and employees available in the customer user registry that WebSphere Portal is configured to use, whether that is an LDAP user registry or a custom registry. The Organization view is used to help users filter the data generated by the Report view based on employees available in the selected organization or organizational unit, or based on a selected employee.
- Export values view: The purpose of this view is the export of an XML representation of the actual values to be used for the feedback loop with the WebSphere Business Modeler.

A launchpad installation helps you effectively deploy WebSphere Business Monitor, which includes a limited license for specific components:

- DB2 UDB Enterprise Server Edition
- DWE OLAP
- DB2 Alphablox
- WebSphere Portal
- WebSphere Process Server

> **Note:** For instructions about installing WebSphere Business Monitor, see the ITSO Redbook, *Business Process Management: Modeling through Monitoring Using WebSphere V6 Products*, SG24-7148.

6.6 WebSphere Process Server and Integration Developer

Based on SOA and as a single, simplified programming model, WebSphere Process Server V6.0 is a business process server that delivers and supports all styles of integration based on open standards to automate business processes that span people, applications, systems, platforms, and architectures.

WebSphere Integration Developer V6.0 is based on Eclipse technology and is a tool for rapid assembly of business solutions that allows you to describe all styles of processes with a programming model based on BPEL. However, if you are going to monitor your processes, you should start with WebSphere Business Modeler and then export to WebSphere Integration Developer for assembly of the executable process. Recall that WebSphere Business Modeler is where the process gets the Business Measures Model.

Benefits of WebSphere Process Server V6.0 include:

- eService component architecture: A simplified integration framework that leverages existing IT.
- Describes processes: Visual editors for component development, assembly, integrated testing, and deployment.
- Support for all styles of integration: Including human tasks, roles-based task assignments, and multilevel escalation. Visual editors for component assembly.
- Change business processes dynamically: Caution is advised here because business processes must still match the business measures model, which comes from WebSphere Business Modeler.
- Dynamically choose implementations: Business rules, business state machines, and selectors dynamically choose implementations for a specific interface based on business scenarios.
- Broadest reach in integration: Built on Enterprise Service Bus (ESB) technologies and support for IBM WebSphere Adapters.
- Business-to-business (B2B) support: Support for B2B through a restricted use license of IBM WebSphere Partner Gateway.

IBM WebSphere Integration Developer V6.0 and IBM WebSphere Process Server V6.0 deliver a composite application platform optimized for building service-oriented applications that extend and integrate a company's existing IT assets.

Note: Some WebSphere product names have changed. For example, WebSphere Process Server V6.0 is the successor to WebSphere Business Integration Server Foundation 5.1.1. WebSphere Integration Developer is the successor to WebSphere Studio Application Developer Integration Edition V5.1.1. And, WebSphere Process Server V6.0 is the successor to WebSphere Business Integration Server V4.3.

WebSphere Process Server V6.0 offers process automation, advanced human workflow, business rules, Application to Application (A2A), and B2B capabilities all on a common, integrated SOA platform with native Java Message Service (JMS) support.

WebSphere Process Server builds on WebSphere Application Server to provide a Java 2 Platform Enterprise Edition (J2EE) and Web services technology-based application platform for deploying enterprise Web services solutions for dynamic On Demand Business.

WebSphere Process Server includes all of the features available in WebSphere Application Server Network Deployment V6, including J2EE 1.4 support, Web Services Gateway, IBM Tivoli Performance Viewer, clustering, and workload management support.

Included in these products are several complementary products for use in the WebSphere Process Server environment, including restricted use licenses for DB2 Universal Database (UDB) Enterprise Edition, Tivoli Directory Server, and Tivoli Access Manager.

WebSphere Process Server also includes a restricted use license of WebSphere Partner Gateway Advanced Edition to provide consolidated partner services for process integration with the WebSphere software platform. B2B gateway consolidation centralizes B2B communications with trading partner communities, providing a central point of control for interactions among partners and providing a security-rich environment at the edge of the enterprise. B2B gateway consolidation is of particular value when multiple business units interact with the same partners or partners with similar processes.

WebSphere Partner Gateway combines extensive partner profile management capabilities with a simple, reliable, and secure exchange for B2B messages capable of serving multiple B2B protocols and standards to efficiently integrate your processes with those of your business partner community.

WebSphere Integration Developer V6.0, optimized for developing composite applications that deploy to WebSphere Process Server V6, delivers an application development environment for building service-oriented, component-based applications that extend and integrate your IT assets.

WebSphere Integration Developer V6.0 focuses on developer productivity by providing authoring tools that allow integration developers to build and debug composite business integration applications. Combined with other development tools from IBM, for example, Rational Application Developer and WebSphere Business Modeler, it provides support for enterprise developers.

6.6.1 Process server and integration developer - together

Together WebSphere Process Server V6.0 and WebSphere Integration Developer V6.0 provide a list of services to enable the development of composite integration applications. These service components include:

- Business processes
- Human tasks
- Business state machines
- Business rules
- Supporting components

Business processes

The business processes component in WebSphere Process Server implements a WebSphere-BPEL compliant process engine. It represents the fourth release of a business process choreography engine on top of the WebSphere Application Server.

WebSphere-BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. Support for WebSphere-BPEL includes:

- A business process editor with authoring experience
- Drag-and-drop tools to visually define the sequence and flow of WebSphere-BPEL business processes
- A visual business process debugger to step through and debug WebSphere-BPEL business processes
- Long-running and short-running business processes
- Compensation support to provide transaction rollback-like support for loosely coupled business processes that cannot be undone programmatically by the application server
- Integrated fault handling to provide an easy and integrated means of performing in-flow exception handling
- Support for including Java snippets and artifacts as part of a business process

Note that we still recommend that you start with WebSphere Business Modeler for creating and editing the business processes.

Human tasks

Human task support expands the reach of WebSphere-BPEL to include activities requiring human interaction as steps in an automated business process. Business processes involving human interaction are interruptible and persistent (a person might take a long time to complete the task) and resume when the person completes the task.

Human task support includes:

- Staff activity nodes to represent a step in a business process that is performed manually
- Java Server Faces (JSF) components to create custom clients
- Dynamically setting duration and calendar attributes for staff activities
- Dynamically setting staff assignments via custom attributes
- Originating task support to invoke any kind of service (including a business process)
- Administrative tasks

You can use human tasks to invoke services (for example a business process), participate in a business process (traditional Staff Activity), or administer a business process (process administrator). Additionally, pure Human Tasks are available to implement ad hoc processing. By separating human task support from the core WebSphere-BPEL engine WebSphere Process Server and WebSphere Integration Developer, you now allow creation of pure WebSphere-BPEL code without IBM extensions for human tasks.

Business state machine

WebSphere Process Server V6.0 provides a business state machine component that can be used to model heavily event-driven business process scenarios. These types of event-oriented scenarios are sometimes hard to model in a WebSphere-BPEL model, but they are very easy to model in a state machine diagram. This state machine is modeled after the Unified Modeling Language (UML) state machine diagrams. The combination of WebSphere-BPEL business processes with business state machines gives WebSphere Process Server V6.0 a unique edge when it comes to business process automation.

Business rules

WebSphere Process Server V6.0 contains a business rule component that provides support for Rule Sets (*If Then* rules) and decision tables. Business rules are grouped into a Rule Group and accessed just like any other component.

WebSphere Process Server V6.0 also provides a Web client with national-language-supported plain text display capabilities to allow spontaneous changes to business rules to be deployed using an intuitive user interface. By separating the business rules component from the individual business process flows, a rule can be managed by the domain expert for that particular business rule. By encapsulating rules as a service component, a rule can be used across multiple processes for maximum business flexibility.

Supporting components

WebSphere Process Server V6.0 provides a wide range of supporting components in order to facilitate component-based application development. Among these are:

- Interface maps: Can be used to convert semantically but not syntactically identical interfaces. These are very beneficial for importing existing services which might have a different interface definition than required. They are also beneficial for implementing a completely canonical integration solution where one component has no knowledge of the implementation details of another component.
- Data maps: Can be used to translate one business object into another, for example, as part of an interface map it is often necessary to translate the arguments of an operation.
- Relationships: Can be used to convert key information to access the same data sets in various back-end systems and keep track of which data sets represent identical data. This enables cross-referencing and federation of heterogeneous business objects across disparate Enterprise Information Systems (EISs). Relationships can be called from a business object map when converting one business object into another to manage the key information. Additionally, lookup relationships can be defined for static data, for example, mapping zip codes into city names.
- Selectors: Can be used to dynamically invoke different interface implementations (components) based on various rules, for example, date. When combined with Interface Maps, you can achieve a great deal of flexibility. A Web Interface is provided to change these selector rules spontaneously, for example, invoking a newly deployed module without requiring redeploying the calling module.

6.6.2 Back-end system connectivity

WebSphere Integration Developer V6.0 provides integrated, open standards-based support for building composite applications, including WebSphere-BPEL business processes that integrate with back-end systems, including:

- Integrated tool support for using J2EE Connector Architecture (J2C) 1.0 and 1.5 resource adapters to access back-end systems
- Tool integration for J2C adapters with tool plug-in extensions (available from IBM and IBM Business Partners)
- J2C 1.5 resource adapter support to leverage WebSphere Adapters
- Support for the entire suite of WebSphere Business Integration Adapters
- Tools for creating services out of J2C resource adapters or WebSphere Business Integration Adapters and including those services as part of an integration application
- Wizards to manage the low-level data handling requirements for J2C resource adapters
- Support for Web services (JSR 109/JAX-RPC-based)
- Support for JMS messaging through the integrated WebSphere messaging resources (with full connectivity to existing WebSphere MQ-based networks)
- Support for calling EJB Session Beans
- Wizards to quickly and simply expose CICS or IMS programs as enterprise services, including the ability to import definitions from COBOL, C structures, CICS basic mapping support (BMS), and IMS Message Format Service (MFS) definitions

WebSphere Process Server V6.0 builds on the WebSphere Application Server to provide a J2EE and Web services technology-based application platform for deploying enterprise Web services solutions for dynamic e-business on demand.

6.7 WebSphere Advanced Enterprise Service Bus

WebSphere Message Broker V6 delivers the advanced enterprise service bus. That bus integrates different applications and systems by providing transformation and enrichment of in-flight information to provide a level of intermediation between applications that use different message and data structures and formats. The product enables applications to work together so that they exchange information as if they had been designed to do so from the start. It provides the ability to have a range of connectivity options between applications, to meet both the needs of the applications and the requirement for the distribution of the integrated data, and provides all this within a comprehensive environment separate from the application development. Thus, users such as application developers can concentrate on business logic without reducing application and business flexibility by custom coding connectivity and integration logic in the applications and services throughout the business.

WebSphere Message Broker enriches and distributes real-time information from virtually any source of information held in any format through a network of access points or a centralized broker and out to multiple receiving endpoints, each provided with tailored data. This can provide a powerful approach in unifying the IT potential of an organization.

Connectivity to and from WebSphere Message Broker can take advantage of the assured delivery offered by products such as WebSphere MQ or WebSphere Application Server, meaning that transactions take place and complete even after a temporary network failure, so that users and their customers can be confident that information, which is transmitted as messages, is delivered.

With WebSphere Message Broker, users can connect and integrate nearly all types of applications and data to almost any endpoint through a single extensible and configurable environment. The graphical Eclipse-based programming environment of WebSphere Message Broker provides users with a powerful and architected integration layer which helps them to avoid the burden of writing complex bespoke programming as a part of each application or to make use of each data source. Because WebSphere Message Broker provides functions, such as reusable components, adding new connections, applications, or interactions with data is vastly simplified and releases programmers to concentrate on new and changing business requirements.

Connectivity and information sharing between applications for environments that were never designed to work together free users from manual rekeying of data from one application to another. All of an organization's business data is accessible in real time in usable formats to every part of the business. Users can access data faster and respond better to customer needs. Changes to the business can be implemented faster, with fewer errors, and with no disruption to the business applications.

With WebSphere Message Broker, customers can extract valuable data from existing applications and other sources of data and use it elsewhere in their business. They can even access and widely distribute data from applications for which they no longer have the source code. Users do not need to make costly changes to the applications to take a different view of the data they provide.

Users can be more confident that the information they share with their customers is accurate, timely, and (through taking advantage of the reliability and recoverability offered by WebSphere Message Broker) enhanced when used to connect systems using the assured delivery features of WebSphere MQ.

6.7.1 Information distribution

In an SOA, the enterprise service bus (ESB) optimizes information distribution between service requesters and service providers. Each enterprise can deploy its own unique ESB, reflecting how far it has advanced toward becoming an On Demand Business.

As key components of the IBM WebSphere software portfolio, WebSphere MQ and WebSphere Message Broker enable you to begin deploying or widening the deployment of your ESB today.

Some businesses might find that simple messaging-based connectivity between well-matched applications provides the aspects of integration that they require to implement an effective enterprise service bus. Other businesses might find that by extending these capabilities into their wider deployed infrastructure, they realize the value of other parts of the WebSphere software portfolio. As key parts of the IBM WebSphere software portfolio, WebSphere MQ and WebSphere Message Broker enable deploying ESB.

These programs help maximize the value of your IT investment by broadening the range of environments that this connectivity layer can reach, such as hardware and operating system platforms and non-standards-based programming models as well as J2EE and .NET.

Proven delivery of messages and data between the applications must exist to connect these diverse programming models. The proven delivery mechanisms of WebSphere MQ and WebSphere Application Server can extend the standards-based enterprise service bus with reliable connectivity throughout the enterprise. WebSphere Application Server messaging resources provide a best-of-breed Java Message Service (JMS) implementation for use with J2EE-hosted applications. WebSphere MQ seamlessly extends those messaging resources to non-J2EE environments to integrate virtually anything across more than 80 platforms.

WebSphere Message Broker adds services such as message routing, transformation enrichment, and support for a range of message distribution options and protocols to improve their flexibility and performance. This enables businesses to integrate virtually any applications on any systems, exchanging their data in real time.

WebSphere software provides integration capabilities that enable reaping the benefits of service oriented architectures with connectivity and integration through an enterprise service bus.

6.7.2 Components

WebSphere Message Broker includes four components:

IBM WebSphere Message Broker Toolkit

Microsoft Windows and Linux systems use a broker-specific Eclipse perspective to develop message flows by assembling nodes to route and transform messages. Message flows, message definitions, and any other associated files are packaged into deployment containers called broker archive (BAR) files. An administration perspective enables operations staff to deploy BAR files to any broker within the administrative domain. Administrators can view and control the full operational state of each broker from this perspective.

Broker

The broker is the runtime component where deployed flows operate within containers called *execution groups*, which appear as address spaces for IBM z/OS implementations or operating-system processes for other platforms. Execution groups provide an excellent opportunity for vertical scaling and isolation through the ability to use multiple task-control blocks (TCBs) on z/OS or multiple threads on other platforms, as well as the ability to clone address spaces or processes. An individual execution group becomes capable of using multiple processors offering enhanced scalability when multiple copies of that same execution group are run.

Configuration manager

The *configuration manager* is the single management and administration component for a collection of brokers in a domain. The configuration manager controls all operational actions through access-control lists. It also monitors the broker state and holds topology information related to deployments and inter-broker connectivity. All user toolkits are connected to a configuration manager.

User name server

The *user name server* component is used in publish-subscribe networks to determine the set of users and groups either from the operating system or through a user-defined program or file. These values are sent to both the configuration manager and the broker for subsequent administrative and runtime processing.

6.7.3 WebSphere Message Broker topologies

WebSphere Message Broker Toolkit runs on Linux and Windows systems under the Eclipse environment using IBM Rational Application Developer software. All other components reside on the user's platform of choice, and a broker domain should contain the appropriate mix of platforms to meet the message-processing needs of connected applications. Brokers can be deployed either individually as stand-alone processing engines or in a connected bus to create highly available and scalable architectures.

Whether you use a hub, bus, or arbitrary graph topology is a decision you should make based on your architectural needs rather than on functional characteristics. You can arrange brokers in any topology necessary to meet the needs of your enterprise. You can also deploy message flows and their associated artifacts to one, many, or all of the brokers within a topology. Applications connected to a broker node are able to interoperate with other applications connected to any other broker node within the topology using any protocol or message-format combination. In Figure 6-60, we show how brokers can be arbitrarily connected to create a topology that meets the scalability and availability needs required.

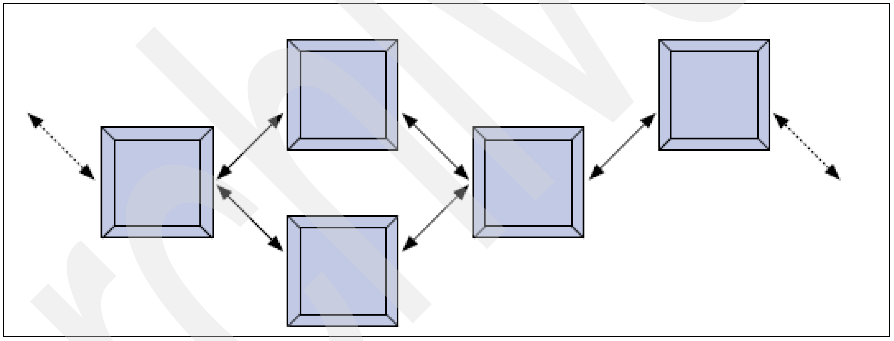


Figure 6-60 Advanced ESB brokers connected

You can also choose to create heterogeneous combinations of brokers on different operating-system platforms. The key requirement behind this need is to be able to link together brokers operating on the appropriate platforms to meet your business needs, and as a result, enable any and all applications throughout your enterprise to be connected. In many cases, this task can involve a combination of z/OS and other platforms, including IBM AIX, Hewlett Packard (HP), Sun Solaris operating environment, Linux, and Microsoft Windows.

Orchestrated reporting and systems

WebSphere Message Broker enables you to collect message-flow accounting and statistics data for an active broker at any time. You can select the granularity of the data that you want to collect by specifying the appropriate parameters in the associated command, and you can view the parameters in force for a broker, an execution group, or an individual message flow. Along with providing deeper insight into how your solution is performing, the accounting and statistics features provide a robust tool for enabling chargebacks in a shared-services environment.

7

Performance insight case study overview

In this chapter, we describe a case study to demonstrate how performance insight can be used in practice. It was developed using a number of the IBM products and technologies that support BIO. We describe all of the products and technologies used in this case study in more detail in other chapters of this redbook. We highlight and use some of the product capabilities, but certainly not all of them. That would be well beyond the scope of this redbook.

In this chapter, we put those products and technologies into action. However, we focused only on the capabilities needed for the case study. But, that in itself is significant because it shows the potential of using the IBM BIO-related products that work effectively together. The case study demonstrates business process management and business intelligence technologies working together to provide performance insight in a typical business context. For more detailed information about BIO-related products, refer to Table 2-1 on page 39 and Chapter 6, “Case study software components” on page 123.

7.1 Introduction

For the case study, we start with a brief description of the business problem that we considered, the Returns Management Process. It is important to understand the case study from a business perspective in addition to the IT perspective. It is the integration of these two areas that provides the strength and benefit.

To begin, we describe some examples of typical work activities in a returns management process, and then we analyze how we gain performance insight to evaluate and optimize them.

In particular, we show how the returns management process, a typical business cross-functional process, can benefit, for example, from programmatic data mining capabilities to make better decisions and influence the process flow itself. We show how to post and resolve alerts about possible process problems, as well as inefficiencies in the process. This is done by monitoring and collecting the results of the operational day-to-day activity and passing them to management and strategic layers for an appropriate evaluation. The strategic layers analyze data accumulated in the process warehouse, and data warehouse from a history perspective, and then analyze that data to find the problem root cause. Action can then be taken to correct the issues, effectively closing the process loop by influencing and optimizing the business performance.

7.2 The returns process management problem

Most organizations implement some form, more or less sophisticated, of a returns management process. In general, customers return products for a variety of reasons, including a simple change of mind, errors, damaged products, wrong quantity, missing items or parts, and so forth.

In some industries, costs for this process can be very high. According to recent surveys, merchandise returns represent 7%-20% of retail sales, depending on merchandise mix and service levels. For example, in the USA, according to the 2003 National Security Survey, retailers lost about $16 Billion to fraudulent returns. It is estimated that on average, 9% of merchandise returns are fraudulent. That means that a retailer with $1B in annual sales could lose as much as $13.5M from fraudulent returns.

Therefore, the Returns Management Process, that spans a number of business functions, could become very expensive for a company. There are many factors that contribute to this. The Returns Process problem cannot be simply reduced to *reverse* the supply chain because there are a large number of peculiarities that differentiate it with respect to processes that constitute the normal business *forward* flow. In many cases, companies experience a scarcity of control in this process, bringing higher costs, low customer service and satisfaction, and inefficiencies.

Return policies vary by industry and supplier and, in some cases, they are also affected by specific legislation. For example, returns can be restricted to a specific period of time after the purchase, and the modality that is used to close the transaction with the customer varies, ranging from a possible refund to a product exchange or to product repair.

Returns provide various challenges to companies because they have impacts both on cost and revenue and, usually, are labor-intensive and information-intensive with, in a number of cases, exceptions to manage. Figure 7-1 shows the main steps of a typical returns process cycle.

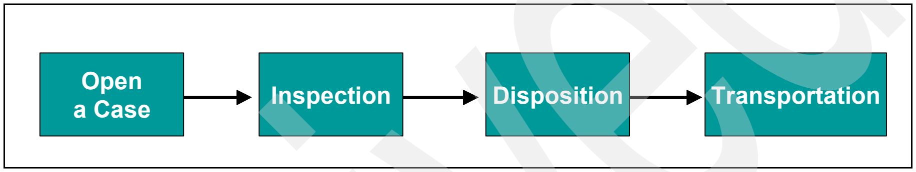


Figure 7-1 Typical returns process cycle steps

A typical returns process cycle can span days or weeks, depending on the industry, to be fully completed. In general, the process consists of four macro-steps:

1. *Open the case*: In this step, the customer contacts the company and issues a request to return a purchased product. The point-of-sale, in-store or via a Call Center representative, is contacted by the customer and a new case is opened. A first check is done to verify if the return is allowed with respect to the policy and customer status. Authorization and a possible refund is asked for and granted to the customer.
2. *Inspection*: After the company receives the product being returned, a number of verifications and inspections are completed to determine the best management strategy to follow on this return.
3. *Disposition*: This activity can be quite complex, because there are several alternative actions that can be taken. For example, the returned product could be repaired. Or it could be exchanged for a new one, and then could either be scrapped, returned to the wholesaler, or sold to a secondary market for subsequent resale. Or a refund could be given to the customer.
4. *Transportation*: In this step, the company is primarily concerned with optimizing the shipping costs associated with the returned product. For example, a new or repaired product could be shipped to the customer, or products could be consolidated and shipped to other destinations (for example, a wholesaler or the vendor).

All these steps can be informative and labor-intensive, and problem tracking can be difficult due to the various consolidation steps that might occur. The amount of information that travels along the process can be significant. For example, it can include a description of what is being returned, information about the customer that is returning the product, reasons for return, the length of time between the purchase and the return, the return authorization code (for example, an RMA number), the returned product value and the amount of credit to refund to the customer, information concerning disposal, and so forth.

Of course, the Returns Process is a cross-functional process in which several roles are acting and several decision points are arranged. Starting from the *Open the Case* step, we can find company representatives operating in the point-of-sale, or at the Call Center, as well as company representatives of the finance department that authorize credit to the customer and send checks or generate other credit forms for the customer. At the *Inspection* step, we can have representatives from the inspection and quality department, packaging, logistics, and so forth. At the *Disposition* step, there are many actors, both technical and commercial, that can be involved in the returns process depending on the specific disposal route that is taken. At the *Transportation* step, representatives from packaging and logistics perform the operations.

The Returns Process has all characteristics required to be a potential candidate for a business improvement target in a typical business scenario, as well as a good problem to analyze through the Performance Insight perspective.

There are several improvement opportunities in this process and these vary, of course, by industry and by the specific company context. In general, the returns process is labor and information intensive and improvements can be obtained from the very beginning. This could be, for example, by designing the process in a way that enables a reduction in the quantity of labor required in each step and by providing all the needed information at the correct time and with appropriate decision support to all actors involved in the cycle.

Finally, the Returns Process problem is also interesting, because we can apply a broad range of evaluation strategies to it. We can apply business key performance indicators (KPIs) related to, and directly influenced by, the process, but also KPIs that evaluate the process itself such as its costs or the time to perform activities. In general, this problem is both a source of a large quantity of data and information and, at the same time, a huge information consumer at various decision layers.

7.3 The case study returns process flow

For the case study, we considered a simplified version of a Returns Process that involves four business roles. Figure 7-2 shows the business roles as well as outlines the macro-process steps that are connected to each business role.

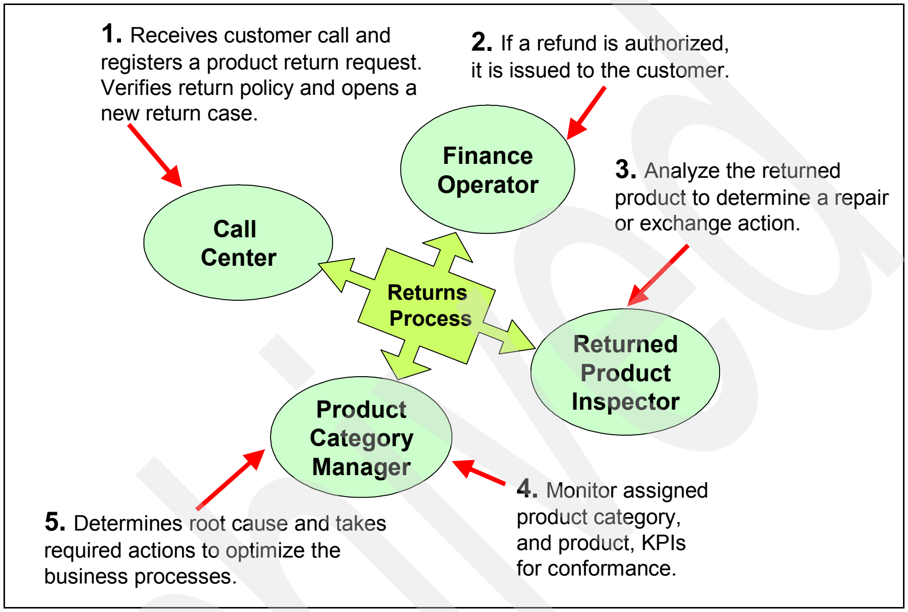


Figure 7-2 The case study return process problem and business roles involved

In this study, we assume that customers interact with the company Call Center and describe their issues to a Call Center representative. The Call Center representative acquires specific details about the customer and about the specific purchase and activates initial verification concerning how the return policy affects this specific purchase. There is also additional information concerning the customer profile and the specific product category itself taken. The Call Center representative comes to an agreement with the customer about the specific return choice that should be followed to close the case. That could be a refund, or, alternatively, to send back to the customer a new product or the original product appropriately repaired.

If a return is determined, a finance department operator is also involved to obtain specific credit authorization to make a refund to the customer.

Once a product is returned to the designated company return center, it is inspected, analyzed, and a specific disposal modality is determined. In this case study, we have defined two disposal modalities. In particular, the decision can be to:

- Repair the product in house. For example, the product has a minor, or no, problem, some items or parts are missing, or a similar condition. The repaired product is then returned to the customer.
- Exchange the product for a new product. Here, the returned product is refurbished and resold as such, or outsourced to a secondary market for subsequent sale.

The Product Category Manager role is responsible to monitor the various processes and events concerning products managed by the company. The Product Category Manager operates at a strategic decision layer. Among other duties, the Product Category Manager verifies the KPIs used with the business processes.

If a KPI exceeds the threshold, the Product Category Manager analyzes the process results and history to discover the root causes of the problems. When a cause is determined, action can be taken to optimize business processes. This, in effect, closes the loop between the strategic and operational layers.

7.4 Performance insight on the returns process

Although the case study covers only a small part of a typical returns process scenario, it has all the elements to provide a good demonstration of attaining performance insight. The returns process is cross-functional and includes various labor intensive steps, as well as a large quantity of information that flows among the various process tasks. Therefore, we could find optimization opportunities from several perspectives.

Additionally, this process supplies a good opportunity to be complemented with analytic activities. For example, it could:

- Bring knowledge inside the process to drive programmatic decisions, and support, in some way, human activities.
- Act as an information source. Process data can flow into the data warehouse to be analyzed and mined, jointly with business data, to discover insights about business performance and formulate appropriate optimization actions.

Figure 7-3 on page 253 depicts a sub-process of the returns process, implemented in the case study, which is managed by the Call Center representative. The process models a set of operational activities corresponding to the interactions that the Call Center representative takes with the customer during the call. It demonstrates how the Call Center operator processes a customer call and how data is provided for performance insight to determine the best solution for this particular customer. For example, data is available on the customer history that can help determine customer value and previous behaviors.

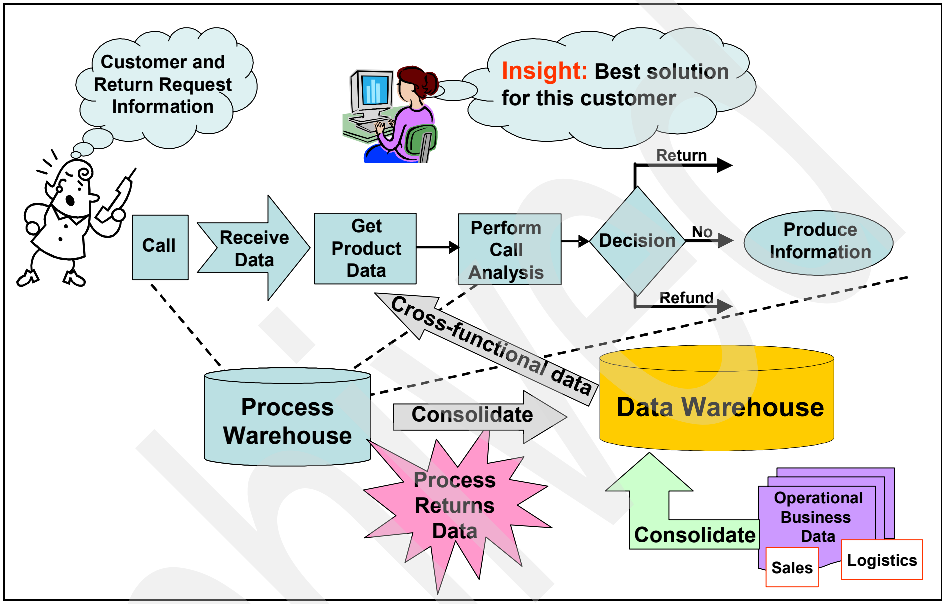


Figure 7-3 The Call Center sub-process

Figure 7-4 on page 254 depicts a sub-process of the returns process that is managed by the Product Category Manager. It includes a number of tasks that are realized, at management and strategic layers, when an alert has been raised about a business performance issue, and a corrective action is taken.

This represents a first analysis concerning process requirements and the business need to introduce performance insights in problem resolution.

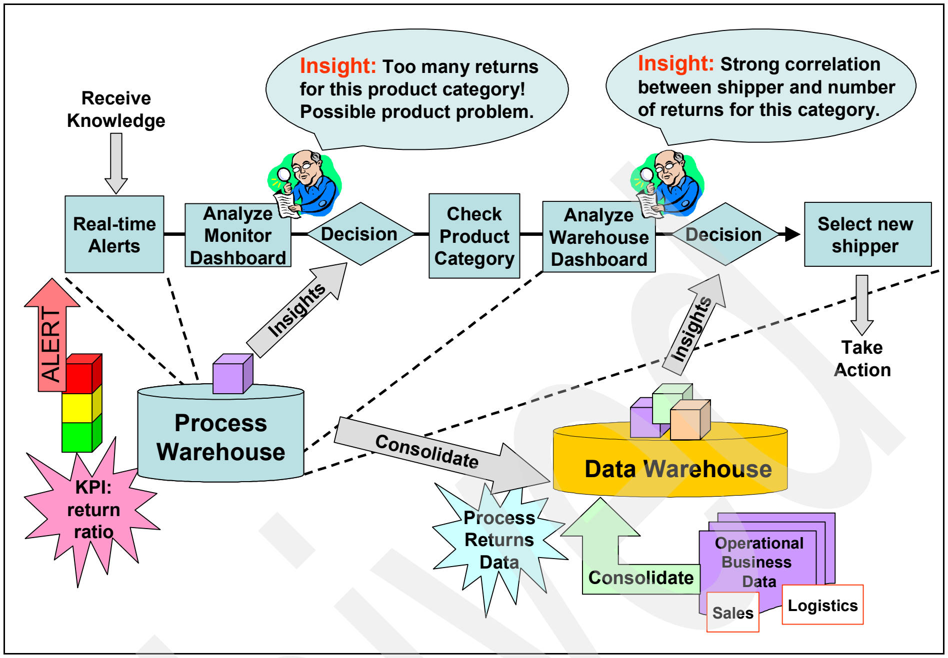


Figure 7-4 The Category Product Manager sub-process

Making a performance insight analysis, in a business context, means to apply the BIO approach introduced in 2.3.1, "An approach to BIO" on page 25. Depending on the complexity of the problem, some, or all, of the steps in the approach must be followed.

To do this requires setting up an infrastructure to support it. You need to be able to model the business processes, and then to run them with both short- and long-running transactions. This process management infrastructure should be able to orchestrate the execution of various business process tasks, both human activities and programmatic activities, and track process data. In the case study, this translates into modeling and running tasks relevant to the Call Center representative and the tasks relevant to the Product Category Manager. Process data, specific process-related KPIs, measures, situations, and so forth are collected by the process management infrastructure and stored in the Process Warehouse.

WebSphere Business Monitor process performance data can be extracted into a company data warehouse to have process history and enable a unified point-of-view relative to business performance. In the case study, this means that significant data concerning the Returns Process is consolidated from the process warehouse into the company data warehouse.

Performance insight is exemplified by the ability to continuously monitor business processes and gain actionable insights revealed through mining and analysis of both process and business data. You need to have an appropriate infrastructure able to aggregate, mine and analyze data, and reveal insights. In the case study, this translates into modeling and development of a specific business measure model and targeting various functional roles, but also in defining analytic models that can help with specific activities.

For example, we can reveal insights to support and appropriately drive decisions of the Call Center representative during the conversation with the customer. By using these insights, for instance, the Call Center representative can be supported in selecting the best business strategy to cope with the specific customer issue. This requires cross-functional data and analysis for the Call Center to support decisions, such as specific product information and statistics, recommendation, and so forth, as represented in Figure 7-3 on page 253. And, at the management and strategic layer, the Product Category Manager might need support to continuously monitor the business performance with respect to various business processes. In particular, concerning the Returns Process, you need to determine a set of measures that are able to reflect the data and information about the process performance. A number of examples that might come from the process or the business data warehouse are included in the following list:

- M_1: Number of returns for a specific product category during last 24 hours
- M_2: Average sales per day for a specific product category during last week
- M_3: Number of returned items from deliveries by a specific shipper
- M_4: Time and cost of each activity of the Return Process (Call Center Activities, Inspections, Transportation, and Finance)
- M_5: Number of refunds in a specific period for specific product categories
- M_6: Number of exchanges in a specific period for specific product categories
- M_7: Number of repairs in a specific period for specific product categories

We can combine measures to form a number of KPIs and associate to each a specific business target. For example, consider the following KPIs and associated targets:

- KPI_1: Returns Ratio, defined as M_1/M_2
- KPI_2: Returns Process Cost with respect to the Total Product Costs
- KPI_3: Percentage of the Returns Process Shipping Cost with respect to the Total Shipping Cost
- KPI_4: Percentage of the refund for specific product categories, defined as $M_5 * 100 / (M_5+M_6+M_7)$
- KPI_5: Customer Reject Ratio, which measures the ratio between the number of products and product parts delivered to customers and the ones returned

WebSphere Business Monitor can generate alerts for target violations on specific KPIs. For example, in Figure 7-4 on page 254, an alert on the Return Ratio KPI is shown. The Product Category Manager receives the alert, generated because the number of returns is becoming too high.

Then, we can go deeper into the investigation and support the Product Category Manager in discovering insight into the root cause of the problem. In Figure 7-4 on page 254, for example, the Product Category Manager discovers a strong correlation between returns and a specific shipper. This resulted from a further business intelligence analysis and was realized by correlating specific Returns Process measures and other business data in the Data Warehouse. Finally, the Product Category Manager might need decision support to select the best corrective action, for example, to change the shipper for a specific product category and replace the shipper by selecting the best shipper from a B2B emarketplace.

In summary, performance insight requirements for a business process means:

- Setting up an appropriate infrastructure to model, manage and orchestrate processes
- Designing a business model and setting up an infrastructure to define and continuously monitor business measures and process KPIs with respect to business targets and reveal insights useful to distinguish among alerts and determine, in real time, where problem areas are located
- Designing a model and setting up an infrastructure able to reveal insights, by using data mining and analytic capabilities, and the ability to optimize the business process and support the various functional roles participating in the process

In subsequent chapters, we provide details about the technological implementation of such performance insight solutions for the case study. In particular, all of the steps are proposed, starting from information gathering, to the outline of a first draft solution architecture, to the definition of the process models for the Call Center and Product Category Manager sub-process, to the definition of the Returns Process specific business measure model, and the definition of the business intelligence model able to support the Returns Process decision activities.

In addition, we describe the primary implementation aspects. In particular, you will find the description of the information representation and integration model, the process integration implementation and deployment activities, and the description of the activities concerning the specific performance insight client application that targets the Product Category Manager.

8

Performance insight case study implementation

Well, this is the last, but perhaps most important chapter in the redbook. That is because it is the culmination of the redbook case study. So far, we have:

- Discussed and described business innovation and optimization
- Described and positioned the Performance Insight on-ramp
- Discussed and described business process management
- Discussed and described business intelligence, positioning it with business process management to comprise performance insight
- Identified and discussed some of the key products from IBM that enable BIO and performance insight
- Defined and described the redbook case study on performance insight

In this chapter, we describe and demonstrate the implementation of the performance insight case study solution. As a brief summary, we described the processes, defined them with WebSphere Business Modeler, assembled them in WebSphere Integration Developer, and executed them in the WebSphere Process Server. We monitored their execution with WebSphere Business Monitor, and demonstrated how we gained performance insight to resolve the business problem. There is, of course, more detail, and you will find that in the remainder of this chapter.

8.1 Solution architecture

In this section, we describe the architecture developed for the implementation of a performance insight solution for our case study. The case study concerned the improvement of the product returns process. In particular, we designed and implemented a solution, within an enterprise portal, with a number of performance insight functions targeted at assisting a Product Category Manager.

Figure 8-1 on page 261 shows the high-level component model for the performance insight returns process solution. Specifically, the performance insight client applications consist, in our case, of a number of specialized monitoring, as well as analysis, portlets. From these client components, the Product Category Manager can constantly receive information and alerts from the day-to-day business process activities, investigate initial problem signals by using process monitoring capabilities, and, finally, perform data analysis by using specialized data mining facilities. This enabled a wider and historic perspective to confirm initial insights, find problem root causes, and initiate possible optimization actions.

Back-end components of the solution architecture include a process engine, able to orchestrate and execute business processes, a monitor engine connected with the process, which is able to monitor and identify possible business target violations and generate initial insights, an information integration engine that provides the necessary layers to manage data consolidation as well as access to the data warehouse and, finally, an analytic engine able to generate insights from the monitors performance warehouse and the enterprise data warehouse.

All components are connected to exchange and replicate information and to acquire services. Figure 8-1 on page 261depicts this. In general, a central role in the architecture is played by the process engine that, based on an enterprise service bus, can be connected also to various heterogeneous company systems, such as the ERP and the Call Center systems.

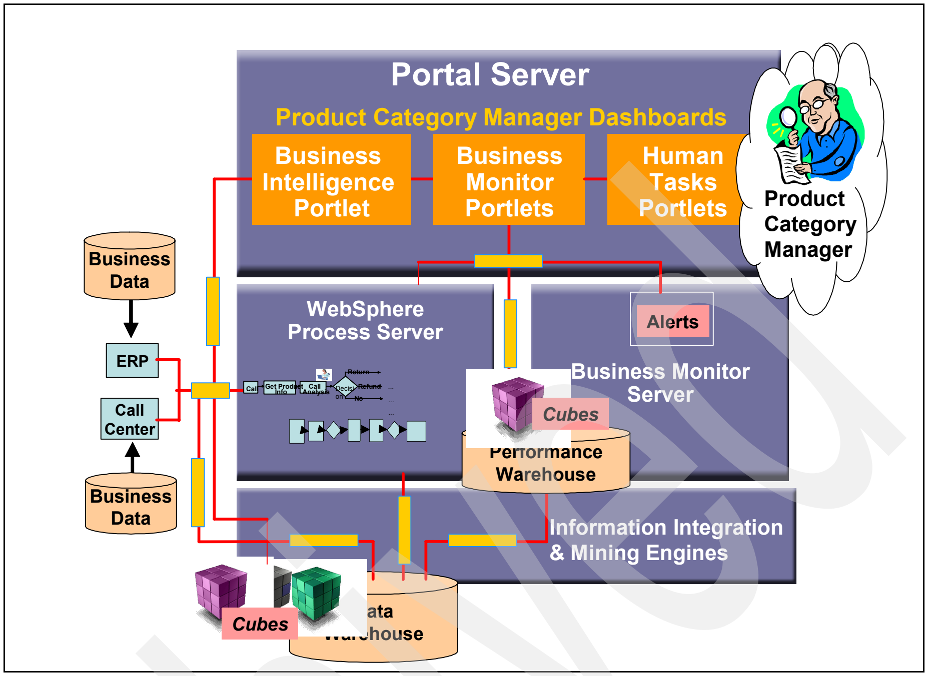


Figure 8-1 The component model

To implement the solution architecture, we used various IBM products (see 2.3.3, “Mapping BIO functionality and IBM products” on page 38). Figure 8-2 on page 262 shows mapping the various products we integrated and the components defined by the model.

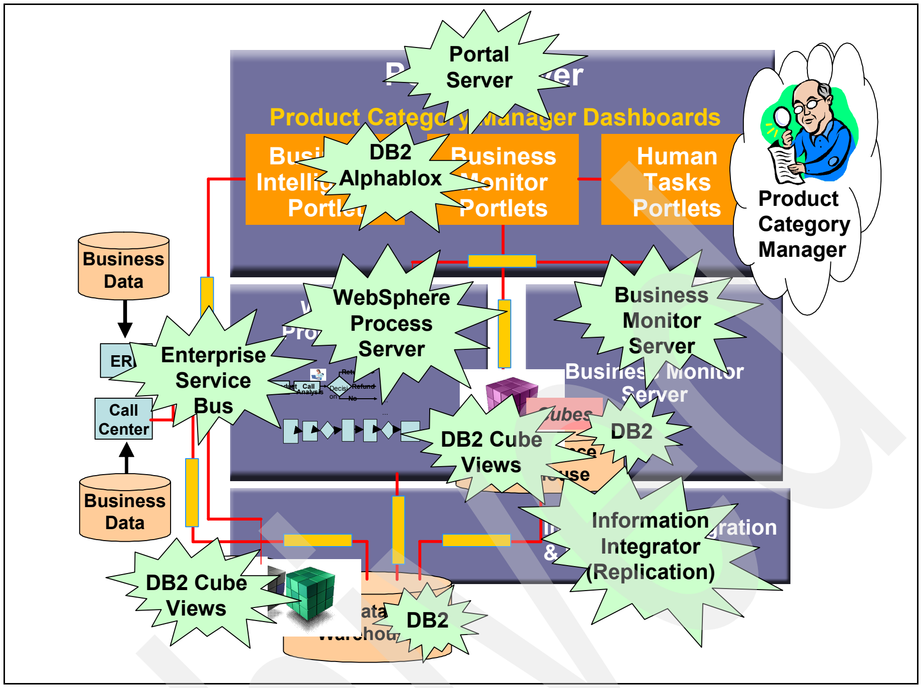


Figure 8-2 Component model and IBM BIO product mapping

The performance insight client application has been implemented as a set of specialized and integrated portlets running on IBM WebSphere Portal Server. Two portlets, Business Intelligence and WebSphere Business Monitor, are implemented by using DB2 Alphablox.

The process engine component, in which processes run, is based on the IBM WebSphere Process Server. The monitoring component is implemented by using IBM WebSphere Business Monitor. Enterprise application and the process tasks are connected through an Enterprise Service Bus.

Operational data and process data are consolidated in the data warehouse. The information integration capability, based on DB2 SQL replication, is used to replicate data from the performance warehouse to the enterprise data warehouse. Both warehouses are implemented using IBM DB2 databases.

Finally, we have simplified analytic content which is in the form of a number of cubes and can be accessed through the client dashboard. In these cases, we used DWE OLAP to generate both the cubes from the performance warehouse that can be accessed from the WebSphere Business Monitor portlet and cubes from the data warehouse that can be accessed from the Business Intelligence portlet. Figure 8-3 shows a high-level operational model for the case study.

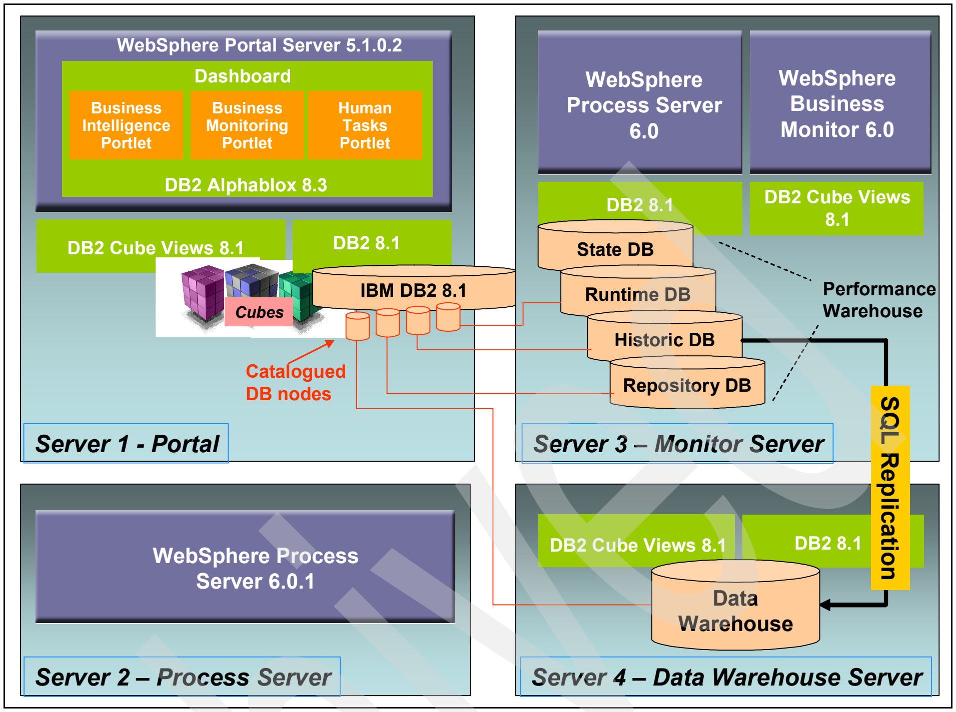


Figure 8-3 High-level case study operational model

That model is based on four servers:

1. Server 1 hosts all client-related components and cubes that are shown through the dashboard. In this example, we have WebSphere Portal Server 5.1.0.2 and three portlets for the Product Category manager. Two of the portlets are for Business Intelligence and the WebSphere Business Monitor and are created by using DB2 Alphablox 8.3, which accesses cubes created using DWE OLAP. The third portlet shows the human tasks assigned to the Product Category Manager. Using this portlet, a task can be claimed and the required corrective action taken. On this server, we also have a DB2 8.1 instance on which is catalogued four databases from the data warehouse and performance warehouse. This is where the cubes are generated.
2. Server 2 hosts WebSphere Process Server 6.0.1, where all the processes are deployed.
3. Server 3 hosts all components related to the process monitoring activity and the first level of insight generation. On this server, we have WebSphere Business Monitor 6.0, which operates on top of a WebSphere Process Server 6.0 instance and receives events from it. The WebSphere Business Monitor is

able to catch events generated from process instances running in the process server.

The performance warehouse consists of four databases (DBs) populated by the WebSphere Business Monitor:

a. The State DB is used by the WebSphere Business Monitor server to store internal and staging data.

b. The Runtime DB hosts information concerning running processes.

c. The Historic DB stores information concerning all completed processes. All these databases are generated programmatically when processes are deployed into the process and WebSphere Business Monitor servers. The data replication between the various databases is managed by the middleware, and the replication schedule is defined by the system administrator.

d. The Repository DB stores schema mapping between the Historic database physical tables and column names, and the names of KPIs and measures reported into the process models by the business analyst. The Historic DB, Runtime DB, and Repository DB are catalogued on the DB2 running on Server 1.

4. Server 4 hosts the data warehouse on a DB2 instance. The tables of the Historic DB database are replicated into the data warehouse database by using DB2 SQL replication feature hosted on Server 3. On this server, an instance of DWE OLAP also operates to generate cubes from the data warehouse.

8.2 Process modeling

In this section, we describe how to create the model and propose a few simple guidelines for the setup of WebSphere Business Modeler.

Before explaining how to create the model, we describe the implementation phases to model, monitor, and improve a process. The phases are indicated by numbered boxes as shown in Figure 8-4 on page 265.

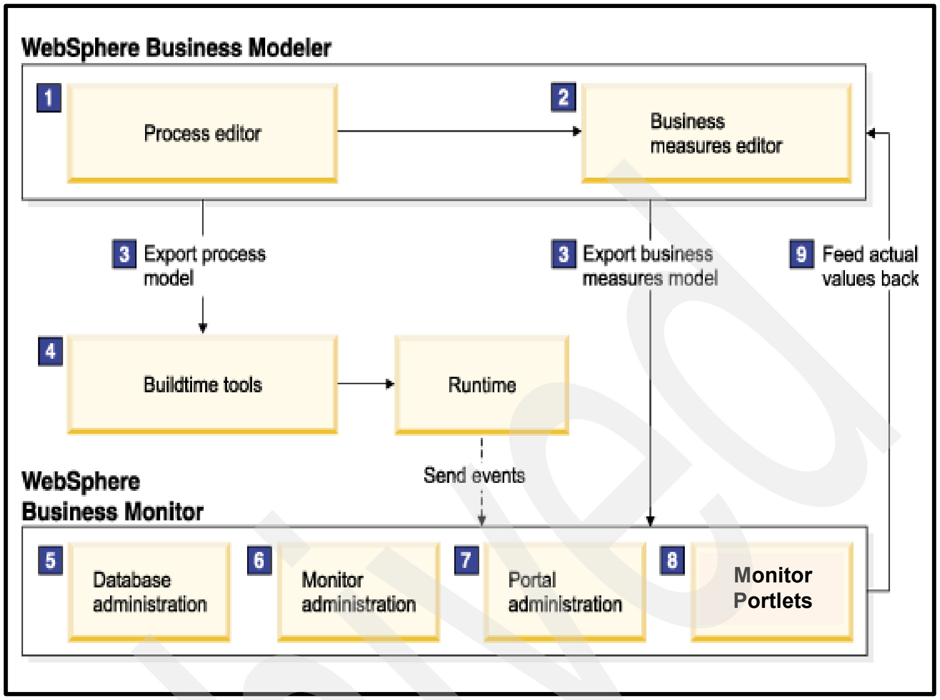


Figure 8-4 WebSphere Business Modeler: Monitor phase

Phase 1: Business process modeling

Business process modeling in WebSphere Business Modeler provides the foundation. A business analyst performs the following tasks:

- Builds and refines the process model
- Simulates *what if?* conditions
- Selects the processes for monitoring
- Determines whether processes are optimum, and how they can be measured

Phase 2: Adding business measures

After modeling or importing the process, the business analyst uses the Business Measures Editor to add business measures.

Phase 3: Exporting the model

The solution architect adds the technology-specific information to the model, including the triggers, calculations, and database schema information. The solution architect then exports the model. When the Business Measures Model is exported, the process model is programmatically exported as well.

Phase 4: Automating the model

An IT specialist imports the model into WebSphere Integration Developer, and defines, refines, and implements it. The IT specialist then deploys the model to the WebSphere Process Server.

Phase 5: Configuring the databases

The database administrator imports the Business Measures Model and generates artifacts from the schema generator using the WebSphere Business Monitor administrative console. WebSphere Business Monitor uses databases to store information related to business measures and event definitions.

Phase 6: Importing the Business Measures Model

The administrator imports the results of the schema generation, which include the Business Measures Model and DWE OLAP definitions, into the WebSphere Business Monitor administrative console and performs any required administration.

Phase 7: Setting up the dashboards

The administrator copies and configures each of the portlet views, based on what the business user wants to monitor.

Phase 8: Monitoring the process

Using portlets, WebSphere Business Monitor provides external visibility into what is occurring when the business process is executing. There are also portlets that allow you to examine the historical performance. Based on the Business Measures Model, the monitor receives events, updates metrics, counters, and stopwatches, decides when situations occur, and emits secondary events to report them. As each process instance ends, the KPIs and historical metrics are updated.

Phase 9: Feeding values back to improve the process

Once the process model has been executing for some time, the resulting values can be exported to an XML file and imported back into WebSphere Business Modeler to validate the solution or to perform further analysis on the process.

8.2.1 WebSphere Business Modeler - Getting started

Business modeling is an interactive process, requiring the business analyst to continually revise the process as a deeper understanding of the goals, requirements, and individual activities involved are known. The business analyst must continue to meet with subject matter experts to gather information and validate the draft model.

In this redbook, we describe the steps for business process management, and then generate a model in WebSphere Business Modeler. It can be exported to WebSphere Integration Developer and WebSphere Process Server. What does that mean?

WebSphere Business Modeler provides a number of business modeling modes, each of which offers a different view of the models you create. You can switch between business modeling modes depending on the level of detail for a model or some aspect within it that you want to view. Figure 8-5 is an example of this.

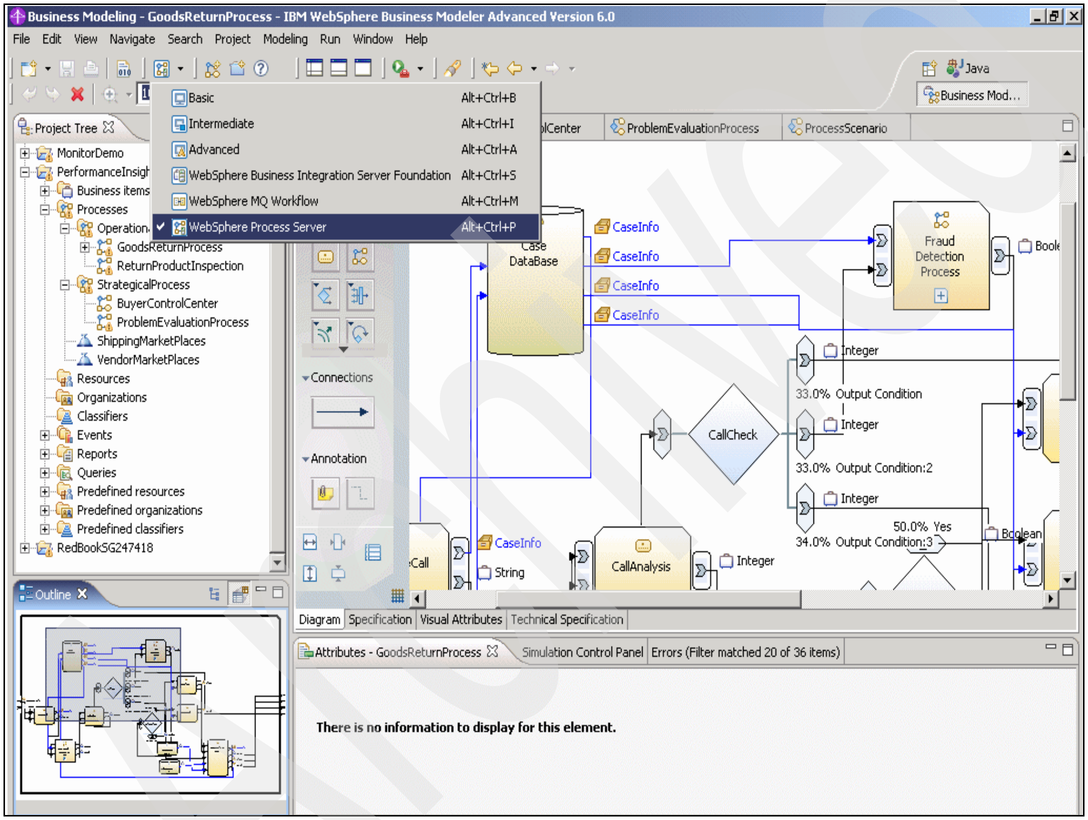


Figure 8-5 Modeling modes

In this case, the WebSphere Process Server mode is optimized for generating output in Business Process Execution Language (BPEL) format. It can then be imported into WebSphere Integration Developer, where you can further define the process for deployment in a runtime environment.

This section describes only the features of WebSphere Business Modeler that were used in the creation of our case study. For more information about the other features, see the WebSphere Business Modeler Help and Tutorial. Select **Help** → **Help Content** to open the help facility. If you are a beginner, expand WebSphere Business Modeler Advance and select **Samples and Tutorials** → **Tutorial Quickstart**. Then go through the complete tutorial to become familiar with the WebSphere Business Modeler.

You can also access WebSphere Business Modeler information through the Web page Help at:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp

8.2.2 Case study implementation

In this section, we present the steps taken to implement the case study.

Creating a project

This is the first step in modeling a new process. A *project* is a grouping of models and other artifacts related to a single work effort. In our case study, the project was called *PerformanceInsight*. Inside the project, we defined all the elements that we needed to implement the study case. In Figure 8-6 on page 269, you can view the elements which we created in this case study. Here, we only explain the element containers (Project, Data Models, Process Models, and Event Definitions) which correspond to the elements created.

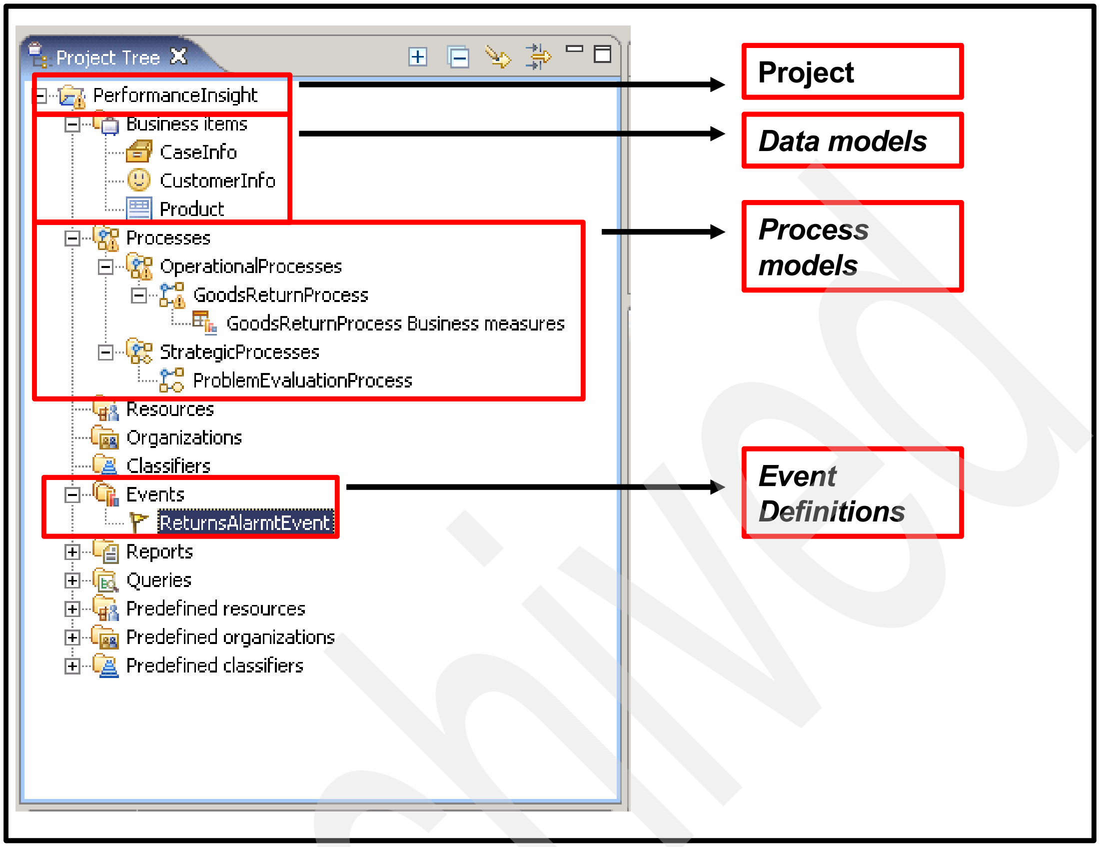


Figure 8-6 Project elements created in our case study

Creating a data model (business items)

The data model includes the business items. These are the documents, work products, or commodities that are used in business operations. In our case, we defined the business items shown in Table 8-1 on page 270.

Table 8-1 Business items defined

Business item name	Business item description
CaseInfo	Contains the *RMAId,* which is the Return Material Authorization Identification number. This ID identifies all the product return transactions. *ProductCategory* represents the product category for items. Example are items such as a Printer or TV. *ShipperName* represents the name of the company that ships the item.
CustomerInfo	Has the *CustId*, which represents the unique customer identification. It also has the customer name, e-mail address, password, and *score* (a representation of the relative importance, or perceived value, of this customer to the enterprise. Is a number in the range 1-10, with 1 being the best rating.
Product	Has the *ProductId,* which represents the unique product identification, the *ProductName,* and *ProductCategory* (which represents a category such as Printer or TV).

As an example, Figure 8-7 on page 271 shows one business item (CaseInfo) developed in WebSphere Business Modeler.

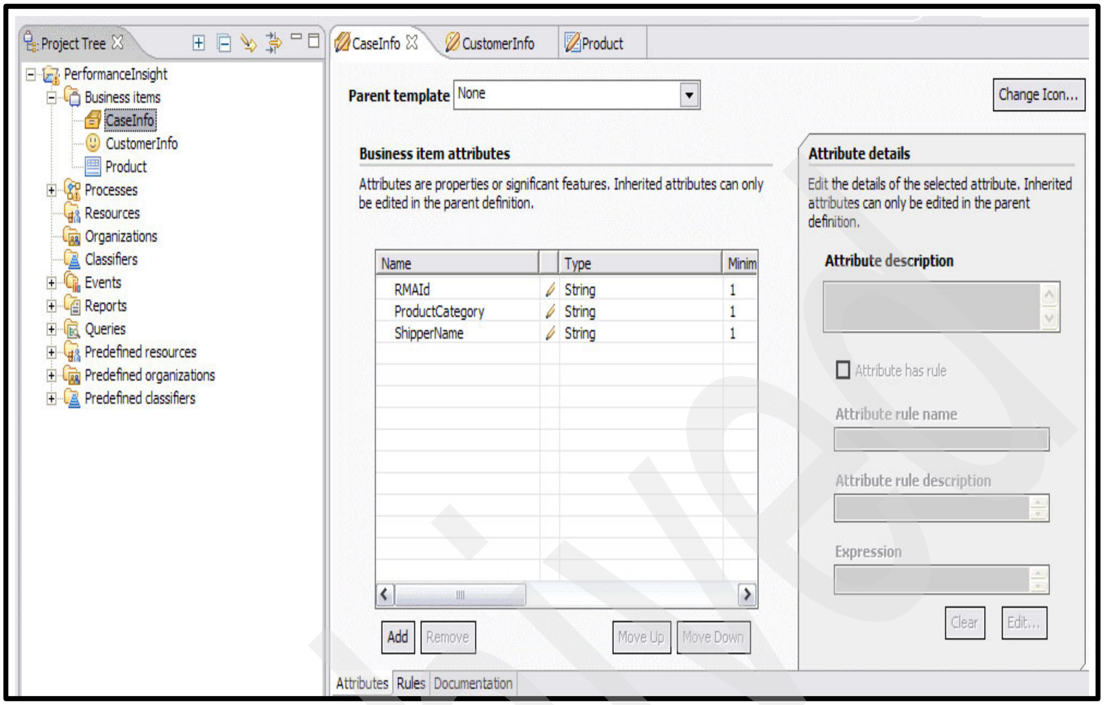


Figure 8-7 Business Item definition

Creating a process

We define two processes, operational and strategic. The operational process is called GoodsReturnProcess and the strategic process is called ProblemEvaluationProcess. Here are the descriptions:

1. **GoodsReturnProcess:** This process represents the starting point in our case study. It begins when a customer calls the company Call Center because there is a problem with a purchased product. During the call, information about the Customer, Product, and Shipper are provided. See Figure 8-8 on page 272 for an example.

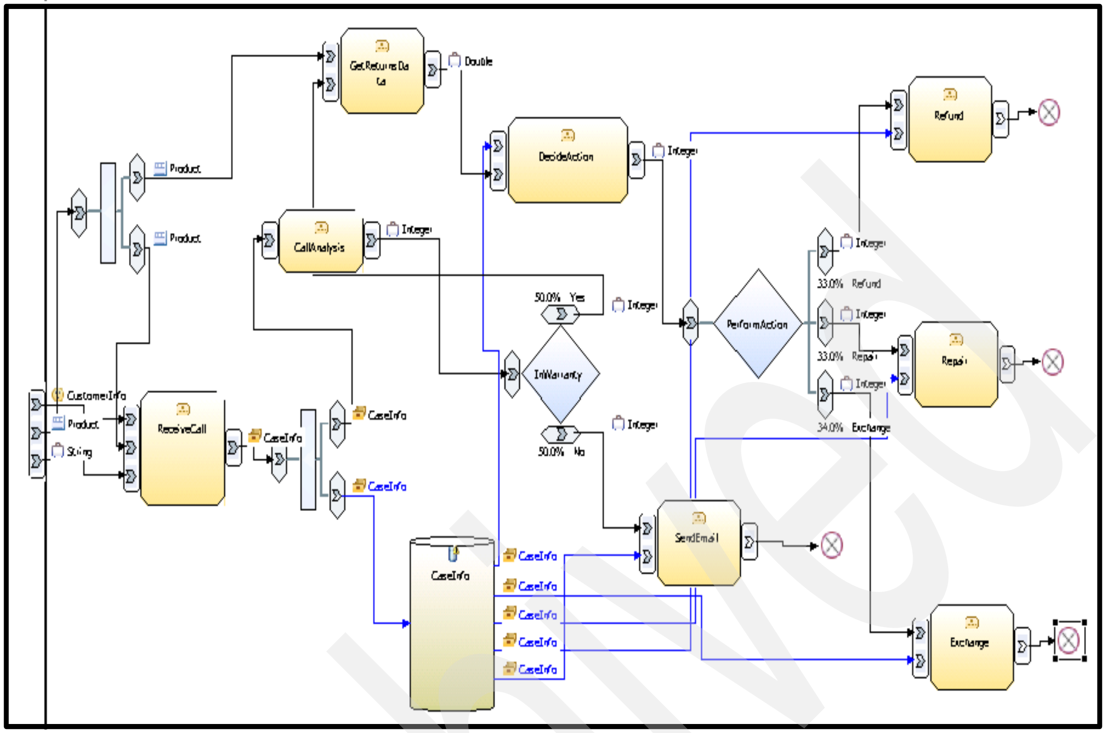


Figure 8-8 GoodsReturnProcess

All the information from the customer is transformed from a modeler point of view in two business items, CustomerInfo and Product. The definitions of these business items are shown in Table 8-1 on page 270. The shipper name is transformed in a string attribute. The first activity that receives this information is ReceiveCall.

To better understand the GoodsReturnProcess, we divided the process into two parts. The first part contains the following elements, ReceiveCall, CallAnalysis, CaseInfo, InWarranty, GetReturnData, and SendEmail. Figure 8-9 on page 273 depicts these elements.

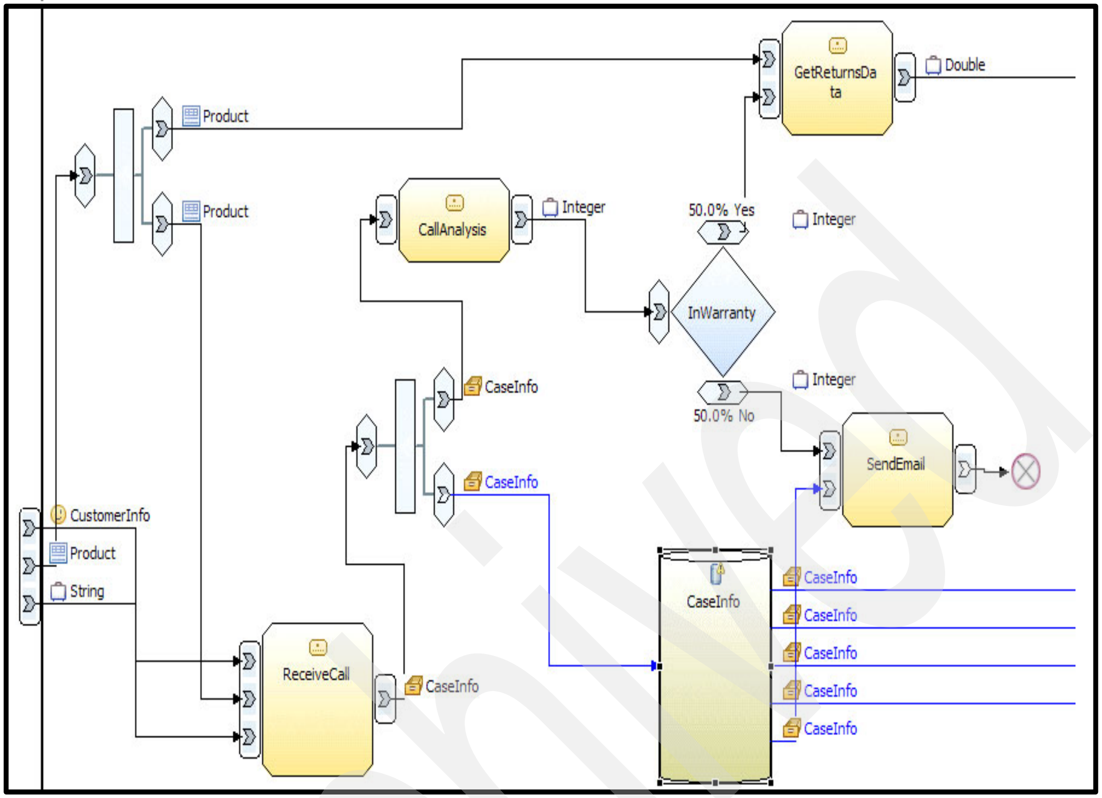


Figure 8-9 GoodsReturnProcess Part 1

The activities in GoodsReturnProcess Part 1 are:

a. **ReceiveCall**: This activity receives the customer information, which is product information and shipper name, and enters it into the Call Center system. The output of this activity is information about the particular case. From a WebSphere Business Modeler point of view, this information is a CaseInfo business item.

b. **CallAnalysis**: Receives the case information and evaluates whether or not the product has a warranty**.**

c. **SendEmail**: An e-mail is then sent to the customer with information regarding a warranty, if there is an e-mail id for that customer.

d. **GetReturnData**: This activity obtains information about the case and specifically about the product. In this activity, we have the combination of business process management and BI, because the system needs help to understand what happened in the business processes execution so a decision can be made.

Part 2 of the GoodsReturnProcess has the elements DecideAction, Refund, Repair, and Exchange, as shown in Figure 8-10.

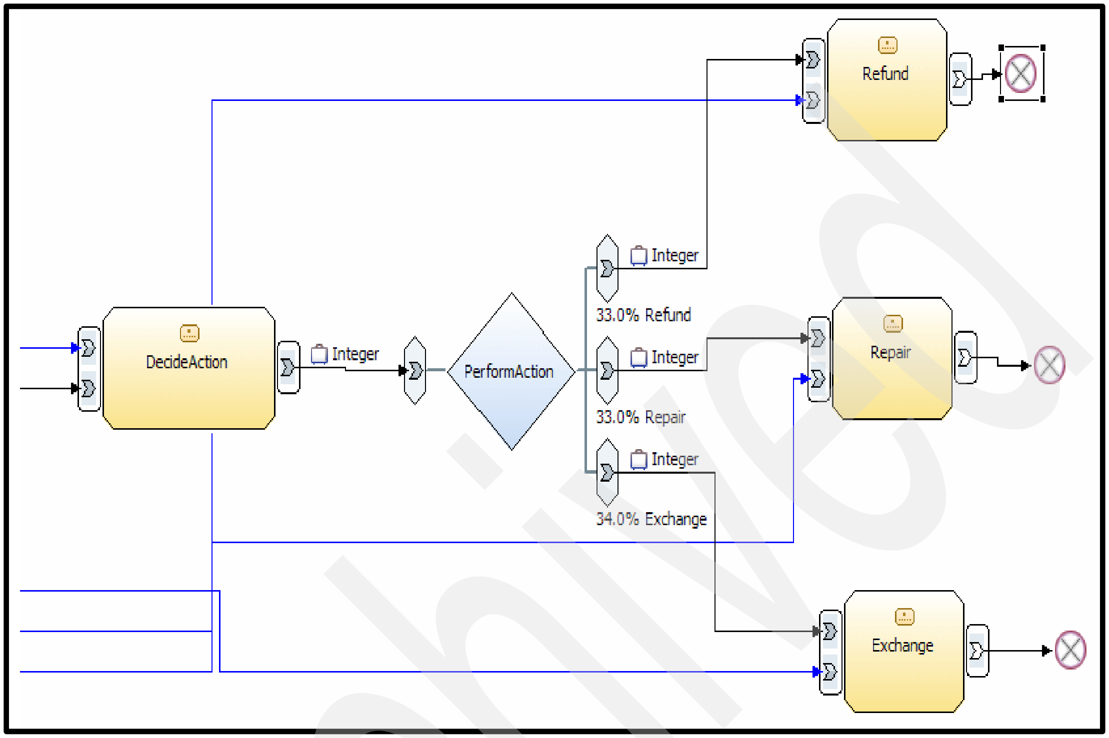


Figure 8-10 GoodsReturnProcess Part 2

The activity in GoodsReturnProcess Part 2 is:

a. **DecideAction**: This activity determines what happened to the product. There are three activity options: give a refund, repair the product, or exchange the product for a new one.

2. **ProblemEvaluationProcess:** Figure 8-12 on page 277 depicts the process as a strategic one. It starts when we have an event or alert in the Enterprise Portal System about this business process. This process has the following activity:

   a. **AnalyzeProblem:** This activity evaluates the problem generated around the business processes execution. The Product Category Manager checks the Enterprise Portal Dashboard to see the event or alert. In this activity, we have the second integration point with business process management and BI. This is required to better understand the event or alert generated to evaluate the business process history by using BI.

When we define this activity in WebSphere Business Modeler, we can define how this activity will be implemented. This particular activity is a human task. A human task is, quite simply, a unit of work done by a human being. Quite often, this task involves interaction with other services and, thus, becomes a task within a larger business goal.

To set this activity as a Human Task, perform the following steps, as depicted in Figure 8-11 on page 276:

i. Double-click the **ProblemEvaluationProcess** in the Project Tree.

ii. On the left side is the Process Editor, where you can see the diagram for this process. Select the **AnalyzeProblem** activity.

iii. Then in the Attribute View, select **Technical Attribute View**. This label only appears when the model is created to export to an IT platform. In this case, we have been working in Process Server Mode.

iv. Go to the Implementation label and select **Human Task** in the Implementation type.

v. Then save the model. **File** → **Save** or Ctrl + S.

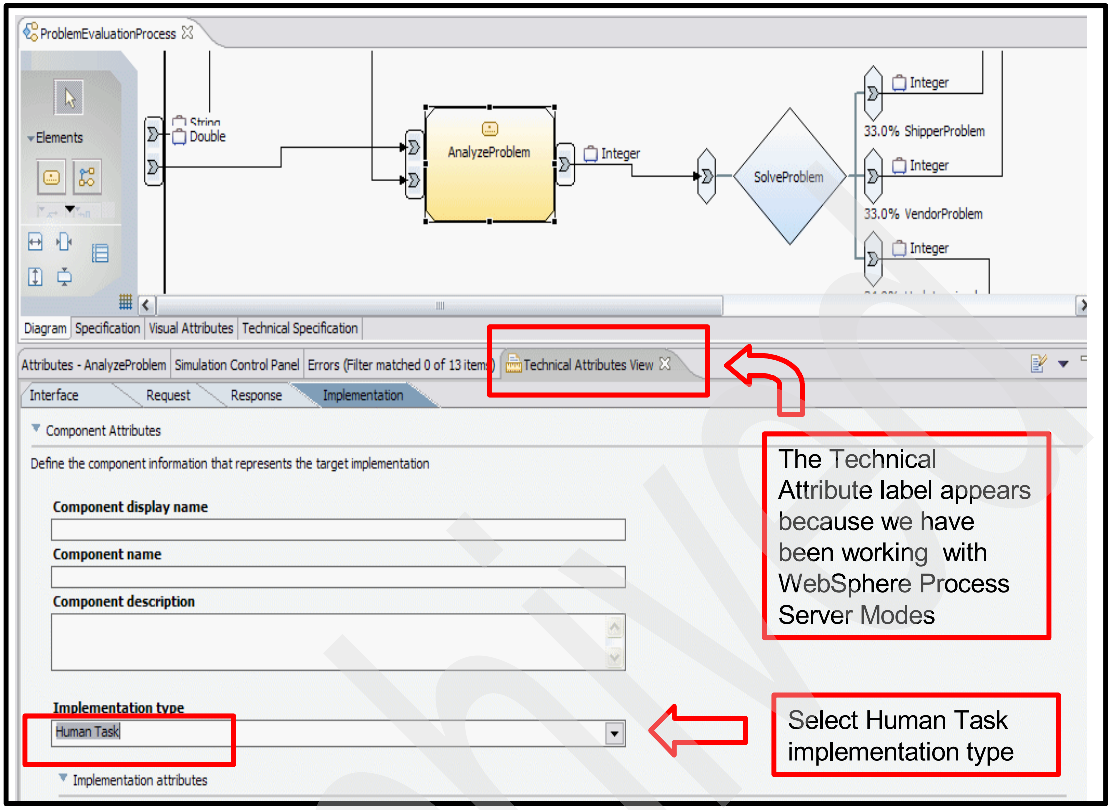


Figure 8-11 Human Task implementation

The following are activity steps:

i. **ChangeShipper**: This activity can change the shipper.

ii. **ChangeVendor**: This activity can change the vendor.

iii. **CallForCSMeeting**: This activity calls for critical situation meeting to evaluate the problem, because it is not common.

Figure 8-12 on page 277 depicts these activity steps.

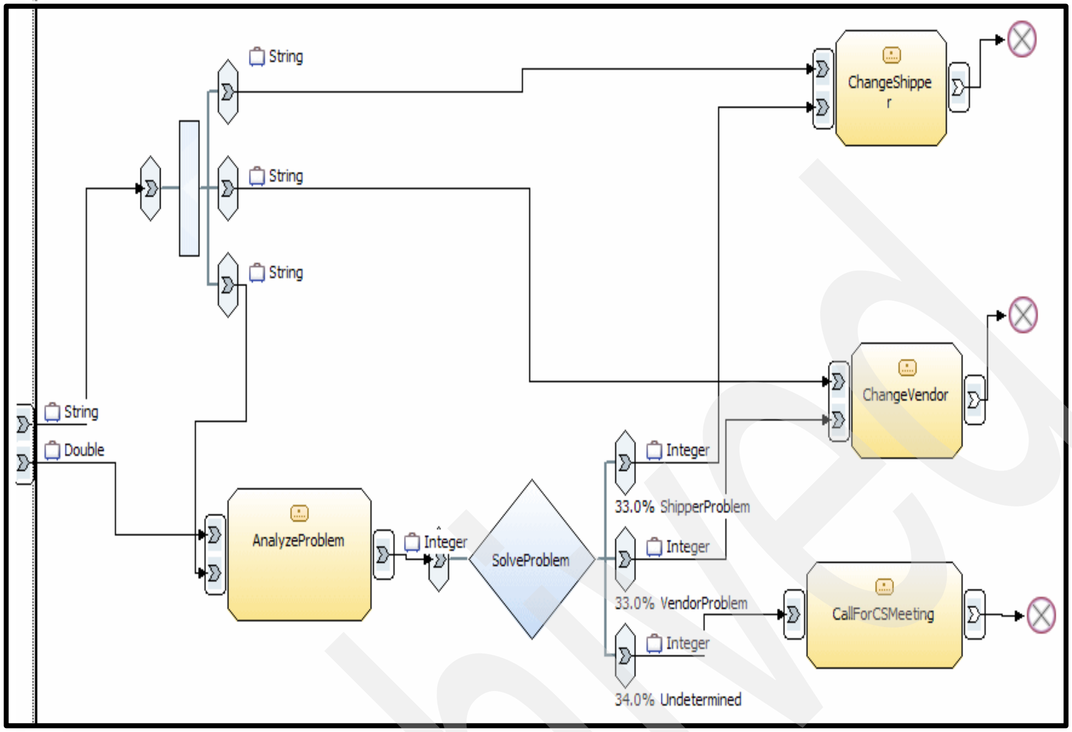


Figure 8-12 ProblemEvaluationProcess

Creating business measures

A Business Measures Model describes business metrics, their dependencies on incoming events, conditions warranting business action (business situations), and situation events that represent notifications of such conditions and might trigger other business actions. Specifically, the Business Measures Model describes how to perform the following actions:

- Gather information from real-time (inbound) events.
- Aggregate information to calculate higher-level business metrics or key performance indicators (KPIs).
- Represent the calculated values on a number of dashboard views and analysis reports, based on the business needs.
- Recognize business situations.
- Emit situation events that can be used to trigger action.

The key to having a successful set of business measures is deciding upon those that are linked to your success. You study the process and the business goals to determine which business measures will be needed from the executing process. When the measures have been established, you can evaluate them in real time and obtain historical information for further analysis.

In our case, we do not use all the elements in the WebSphere Business Modeler Business Measures Editor, we defined metrics, triggers, dimensions and events:

- *Metrics:* A metric is a measurement of a process or process element that is used to assess business performance. A metric can be used alone or in combination with other metrics to define the calculation for a key performance indicator (KPI), which measures performance against a business measurement. A metric is defined within a specific process using WebSphere Business Modeler, and the value of that metric is captured or calculated using WebSphere Business Monitor.
- *Triggers:* A trigger is a mechanism that detects an occurrence and initiates an action in response. For example, you could set a trigger to update a metric each time a task ends.
- *Dimensions:* Dimensions organize data into levels of detail so that you can drill down to extract significant information. Each process can be described in terms of quantitative data, which takes on many values and participates in calculations, and in terms of dimensions, which are entry points for manipulating and analyzing the data in meaningful ways. Generally, any measure with non-numeric values is a level of a dimension, and you analyze other measures against dimensions.
- *Events*: Events are both the source of up-to-date state information from which the values of metrics are derived and the means of notification, which can trigger business actions. The first type are called *inbound events* and the second type are called *situation events*.

Adding a dimension

We can only perform dimensional analysis in the WebSphere Business Monitor if we have at least one dimension specified in the business measures. In our case study, we defined three dimensions, which are:

- ProductCategory
- ShipperName
- ResolutionType

To add a dimension, you can perform the following steps:

1. Select the diagram tab in the Business Measures Model and click into an empty area to see the attributes of the process.
2. Click **Add** in the Dimension section and overtype the default name with, for example, the first dimension, `ProductCategory`. Figure 8-13 on page 279 depicts this.

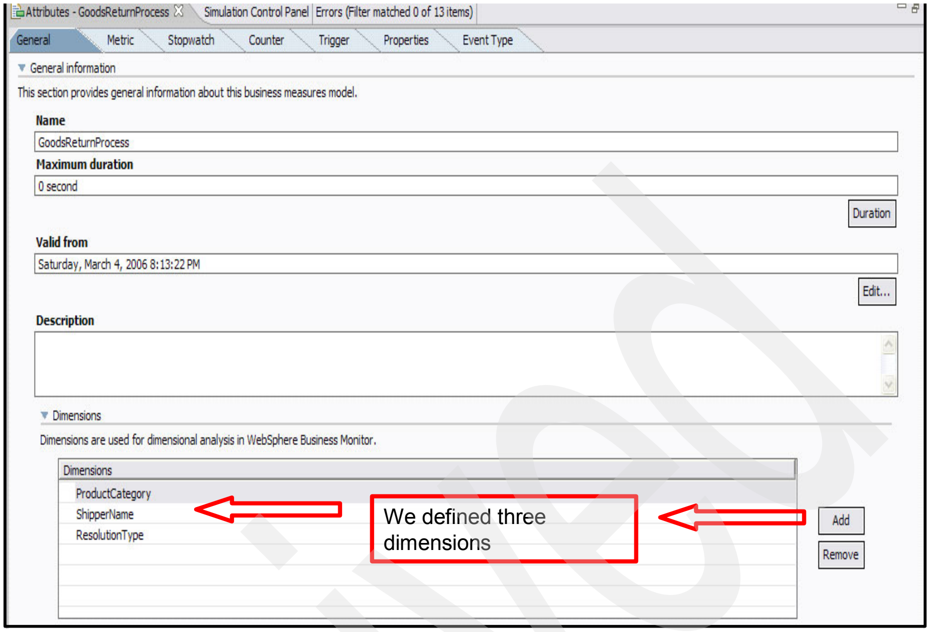


Figure 8-13 Add Dimensions

Metrics

We defined three metrics, which comprise the three dimensions we created, and they are:

- *ResolutionType*: This metric represents which options were selected (NoWarranty, Refund, Repair, or Exchange). It has four triggers associated with each possible option. See Figure 8-14 on page 280.
- *ShipperName*: This metric represents the shipper name.
- *ProductCategory*: This metric represents the product category. For example, DVD or TV.

The last metric represents a value that is calculated for the Product Category Manager, because the metric has an associated event if the value entered when the process is running is over the metric threshold. This metric is:

– *ReturnsRatio*: This metric represents the number of returns for a specific product category during last 24 hours divided by the average sales per day for a specific product category during the last week.

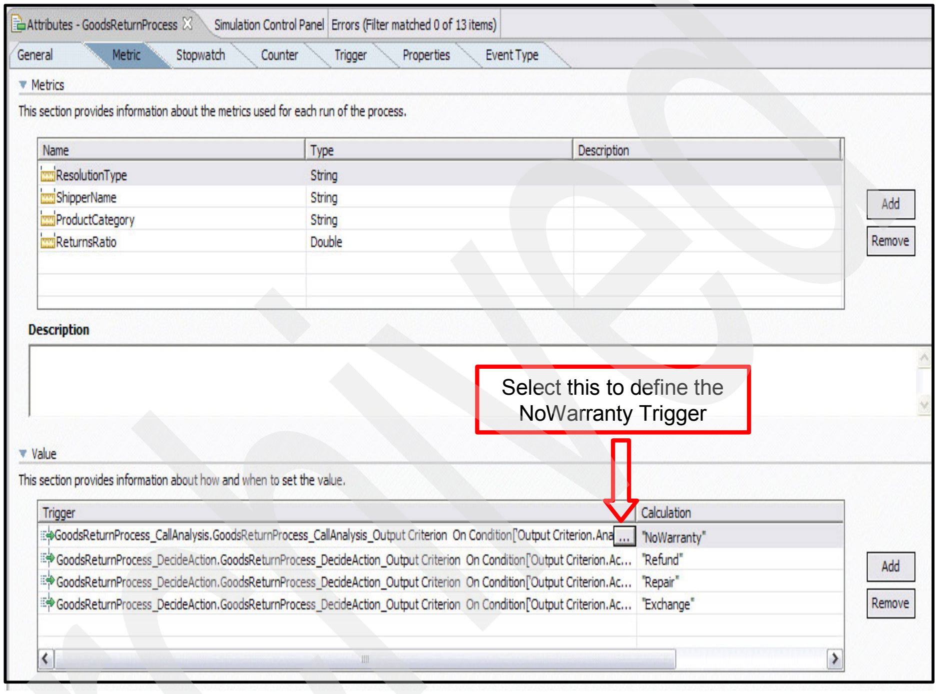


Figure 8-14 ResolutionType metric

To better understand how to use this metric, we provide an explanation under the ResolutionType implementation in the Triggers section.

Triggers

The ResolutionType implementation has four triggers associated with it. These four triggers represent the four possible paths defined for the GoodsReturnProcess, which are NoWarranty, Refund, Repair, and Exchange.

These four triggers have execution conditions, which are:

- *NoWarranty:* The product does not have a warranty. In our case study, the trigger evaluates an integer value, which is 1 for no warranty. The window you use to select and define a trigger condition is depicted in Figure 8-15.

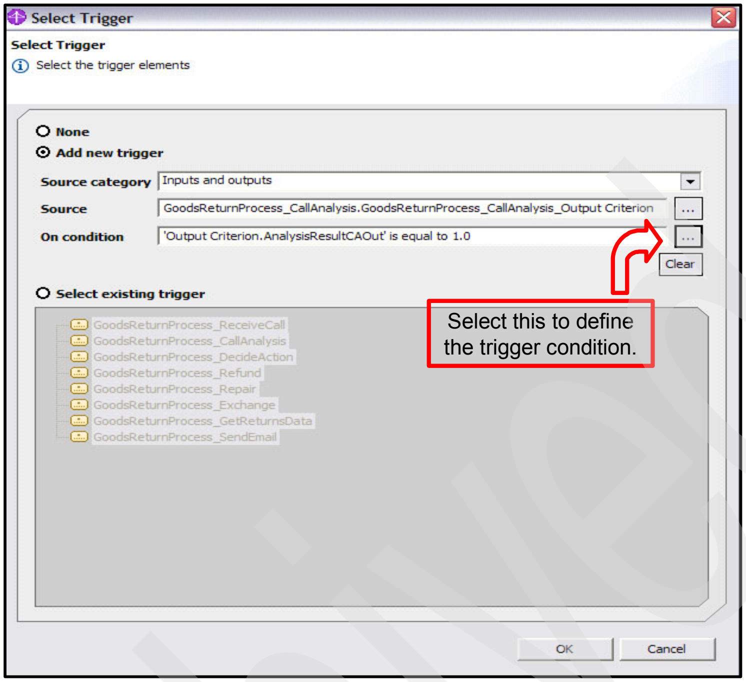


Figure 8-15 NoWarranty trigger

The trigger condition is created with the Expression Builder, which you can see in Figure 8-16 on page 282.

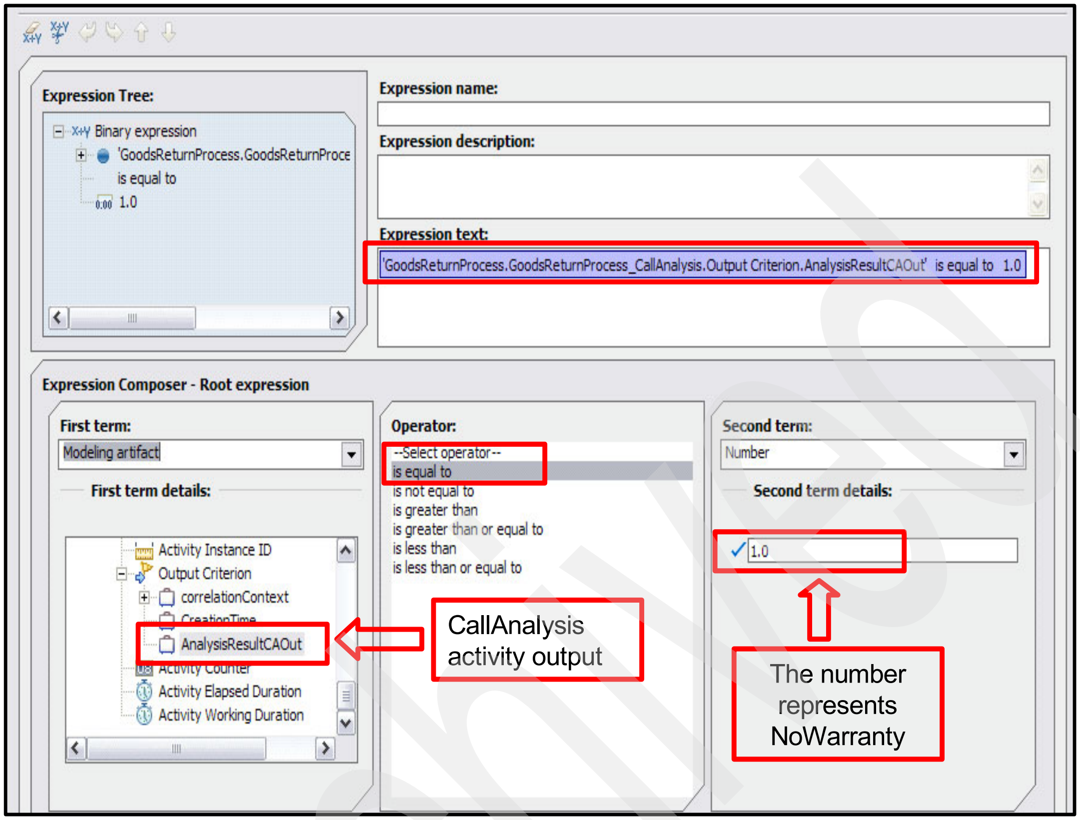


Figure 8-16 No Warranty trigger: Expression Builder

The other three triggers in the ResolutionType metric are:

- *Refund*: The product will be exchanged for a refund. In our case study, the trigger evaluates an integer value. For Refund, the value is `0`.
- *Repair*: The product will be repaired and shipped back to the customer. In our case study, the trigger evaluates an integer value. For Repair, the value is `3`.
- *Exchange*: The product will be exchanged for a new product. In our case study, the trigger evaluates an integer value. For Exchange, the value is `2`.

Adding metrics for dimensional analysis

After you define the metric, you can add the metric for a dimensional analysis. To do that, you can set the dimensional analysis properties:

- Select each metric in turn and select **Aggregation group in dimensional analysis**. Select the maximum length and group level by default value.
- Select **Set as part of the dimension key**, then select the corresponding dimension for each metric. See Figure 8-17.

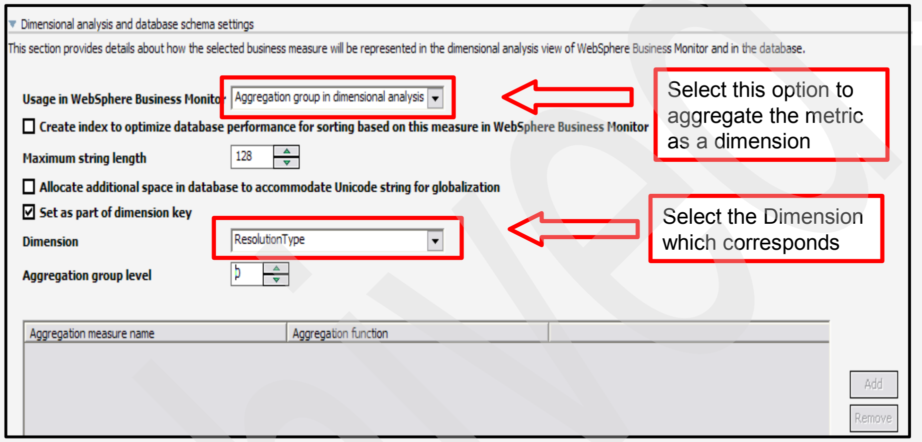


Figure 8-17 Adding metrics for dimensional analysis

Events

We defined the *ReturnAlarmEvent*, and this event has been associated with the ReturnRatio metric. The event is generated when a specific business condition occurs. In this case study, the business condition is that the *ReturnsRatio* is greater than 5%. What does that mean? The ReturnsRatio metric represents the number of returns for a specific product category during last 24 hours divided by the average sales per day for a specific product category during the last week. This result of the division is multiplied by 100 to give a percentage. So when the result of that division is greater than 5 (5%), the event will be executed. An example of a ReturnsAlarmEvent, and how it is associated with metrics, is depicted in Figure 8-18 on page 284.

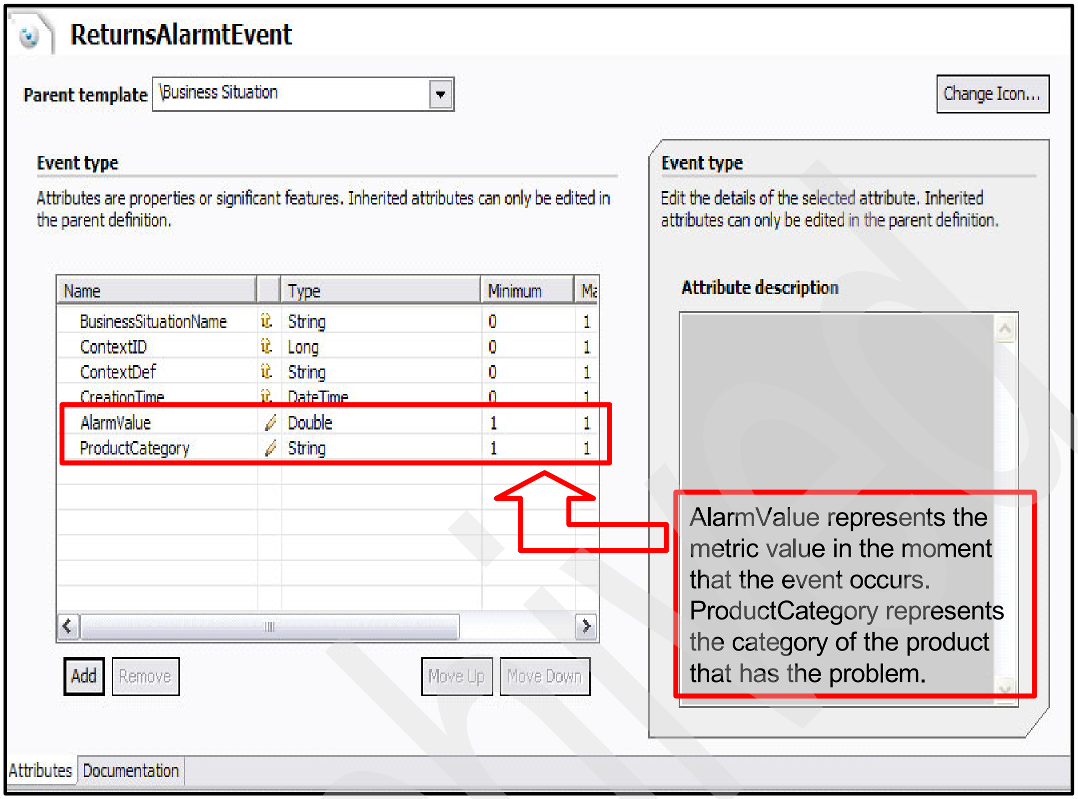


Figure 8-18 ReturnAlarmEvent

After the event was created, we connected this event with the metric ReturnsRatio. Figure 8-19 on page 285 depicts this.

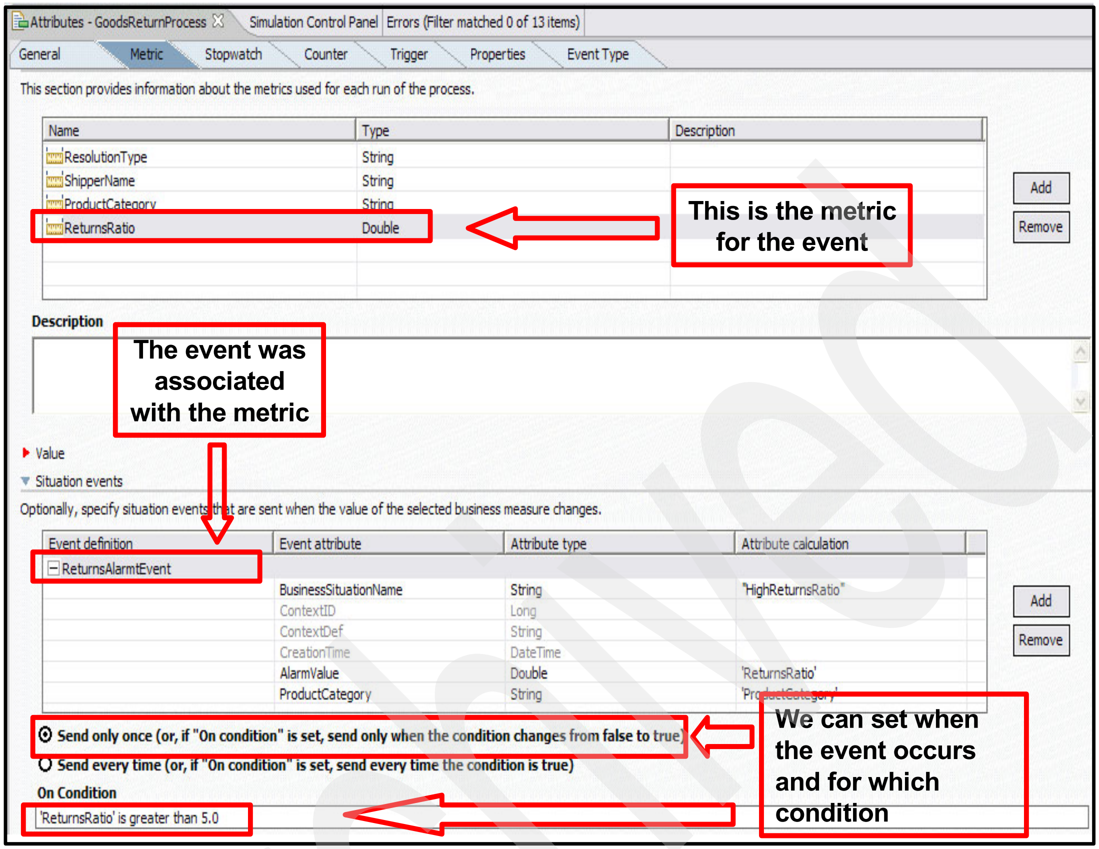


Figure 8-19 ReturnAlarmEvent associated with the metric ReturnRatio

After you associate the event, save the project.

Exporting the project

After completing all the elements in the PerformanceInsight project, the project will be exported to WebSphere Integration Developer and WebSphere Business Monitor.

To do that, right-click over **PerformanceInsight project** and select **Export**. You will see the result as depicted in the example in Figure 8-20 on page 286.

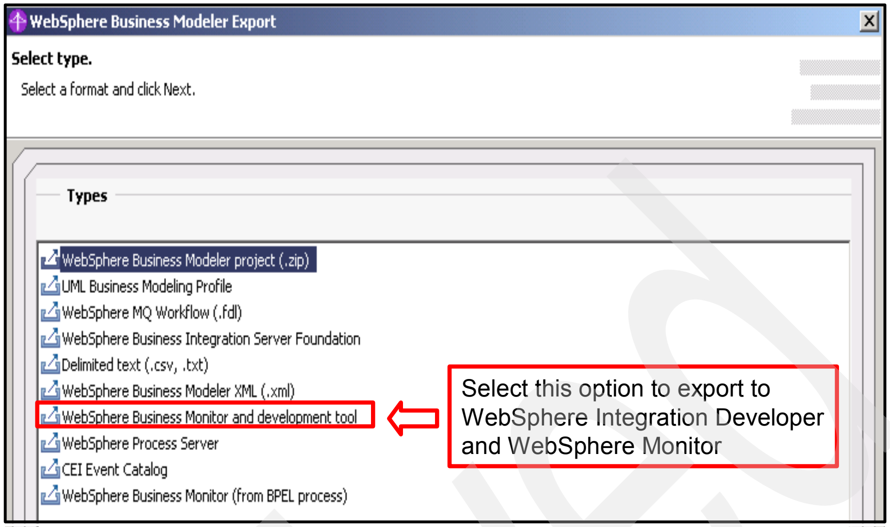


Figure 8-20 Export business model

After you select the type, you can define the following steps:

1. Select **Next** in Select Type window. In the Destination and Source window, you can type the **Target directory**. In this directory, we will have the export files. In **Project,** we have PerformanceInsight.
2. In the next window, select **Module project name** and type `PerformanceInsight` in the text box. In the **Library project name**, type `PerformanceInsightLib` in the text box. Then, select **Project interchange name** and type `PerformanceInsight` in the text box.
3. Then, select **Finish.**

The results of these actions are shown in Figure 8-21 on page 287.

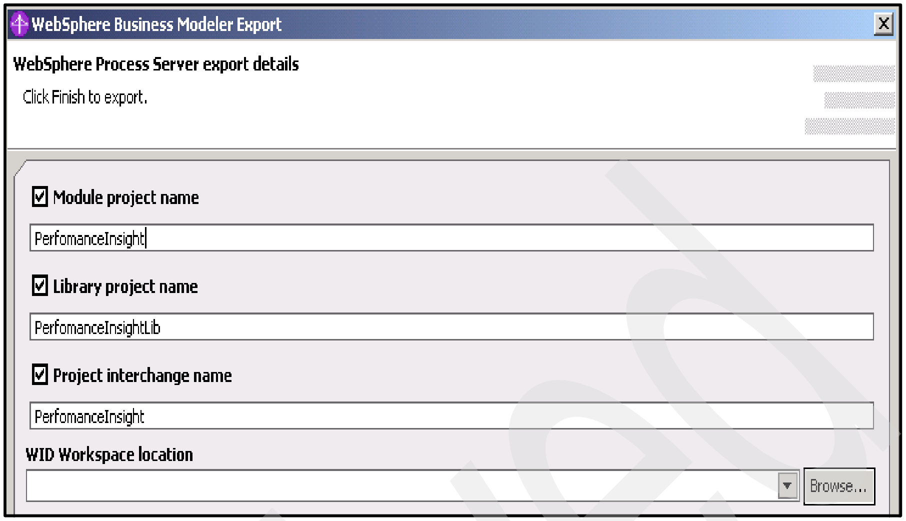


Figure 8-21 Set up model export

We can see two files created: One for WebSphere Business Monitor called Monitor.zip and the other for Integration Developer called PerformanceInsight.zip.

8.3 Process implementation and deployment

So far, we have described the case study architecture, how it was modeled, and the use of business intelligence. Now, in this section, we describe how to get the process up and running. To do that, perform the following steps:

1. Develop the process implementation in WebSphere Integration Developer. Here, you define the real implementation for the process activities, and how they will behave. After finishing the implementation development, export the results to an EAR file in order to deploy the process in WebSphere Process Server.
2. Deploy in WebSphere Process Server. Now, you can deploy the runnable EAR file in WebSphere Process Server.
3. Import the model into WebSphere Business Monitor. To monitor the process and generate alerts, you must import the Monitor.zip file exported from WebSphere Business Modeler into WebSphere Business Monitor.
4. Configure the Adaptive Action Manager. This is to catch the triggered alert and invoke the corrective action Web service. Instantiate a process instance for the ProblemEvaluationProcess, which starts by a Human Task assigned to the Product Category Manager to resolve the problem.

8.3.1 Process development in WebSphere Integration Developer

From an SOA point of view, we will have two modules when we import the project interchange files: PerformanceInsight and PerformanceInsightLib. However, when the Adaptive Action Manager calls the Web service, it sends the Common Business Event (CBE) serialized XML as a parameter to the Web service. So, we need a module in the middle whose sole function is to parse this CBE XML and call the ProblemEvaluationProcess, passing the extracted *ProductCategory* and *AlarmValue*. For an example, see Figure 8-22 on page 289.

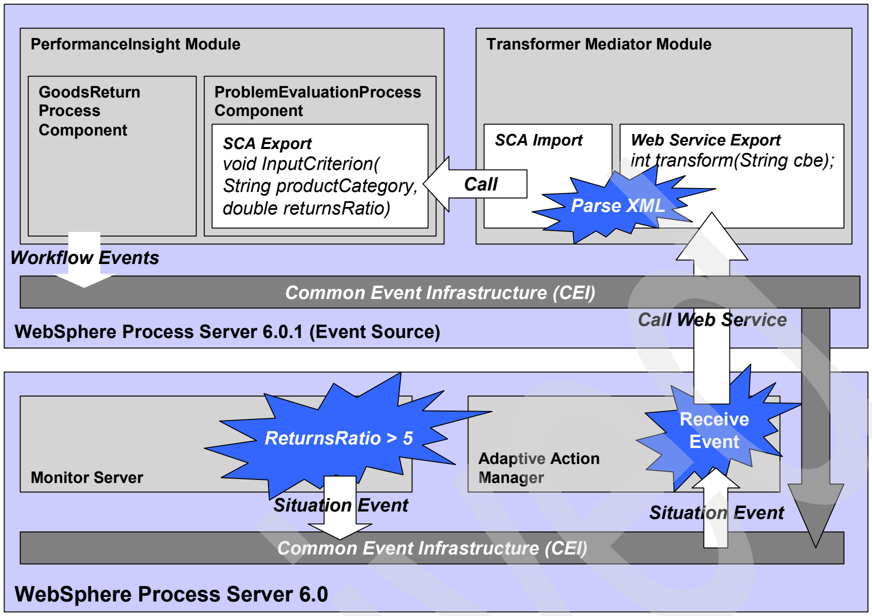


Figure 8-22 Interaction between Service Components and WebSphere Business Monitor

To be able to develop the real process implementation, you will need first to import the Interchange Project file that was exported from WebSphere Business Modeler into WebSphere Integration Developer. To import the file:

1. In the menu bar, click **File**.
2. Click **Import**.
3. Select **Project Interchange.**
4. Browse to where the ZIP file is saved, and check the modules you want to import. In our case, they are PerformanceInsight and PerformanceInsightLib.
5. Click **Finish**.

The project trees for both PerformanceInsight and PerformanceInsightLib modules are shown in Figure 8-23 on page 290.

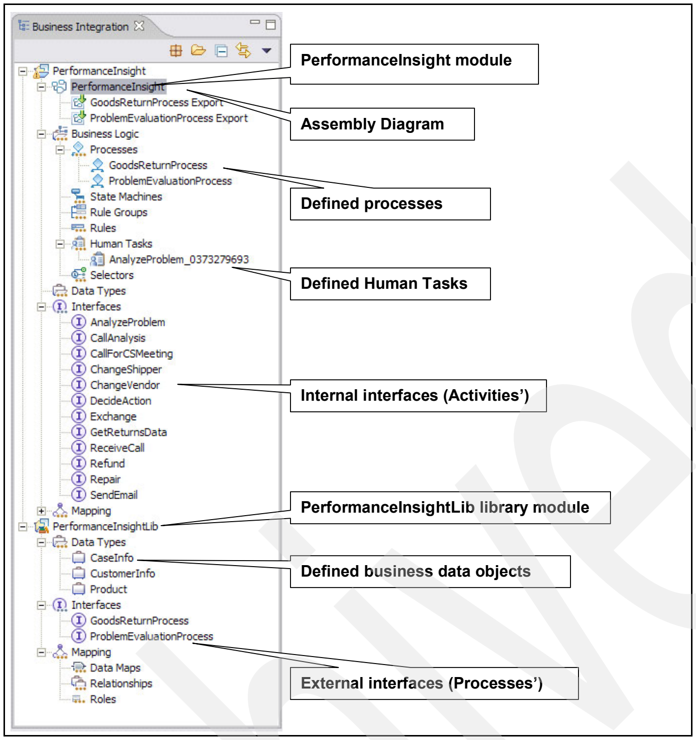


Figure 8-23 Business Integration view in WebSphere Integration Developer

Implementing the GoodsReturnProcess

Open the assembly diagram of PerformanceInsight module (Double-click the ProcessInsight node in the Business Integration view). In GoodsReturnProcess, we have eight activities (Figure 8-24 on page 292 shows the BPEL diagram for GoodsReturnProcess). A service component is generated for each activity, and you have to provide an implementation of these components. The six components and their implementations are depicted in Figure 8-25 on page 293 and the steps are as follows:

1. GetReturnsData: Implemented as Java code, this method calculates the number of returns for the product for the current day relative to the average number of returns in the last week for this product. To implement the component:

   a. Right-click the component in the assembly diagram and click **Generate Implementation.**

   b. Click **Java**.

   c. Choose the package in which you want to add your implementation class.

d. Click **OK**.

e. The Java editor is open to enable you to type the required code.

GetReturnsData retrieves the ratio between number of returns for a product and the average number of sales for the same product during the last week. The sales information extracted from the data warehouse database is responsible to provide the average number of sales per day. This was accomplished by the consolidation of the number of items sold per day considering sales for the last seven days. An MQT was created and is refreshed every day with the latest sales per day.

The source table with sales transactions is called DWH.ITM_TXN. Based on this table, we created a table with sales from the last seven days.

The creation statement for the MQT that hold sales for the last seven days is depicted in Example 8-1:

Example 8-1 MQT creation

```
CREATE TABLE DWH.AVG_SALES_DAY AS (  SELECT PD_ID, COUNT(pd_id) AS COUNT,
DAY(ITM_TXN_TMS) AS DAY, ITM_TXN_TMS AS TMS FROM DWH.ITM_TXN GROUP BY (PD_ID,
ITM_TXN_TMS) HAVING PD_ID=34195 AND JULIAN_DAY(ITM_TXN_TMS) >
JULIAN_DAY(CURRENT TIMESTAMP) - 7)  DATA INITIALLY DEFERRED REFRESH DEFERRED
```

Notice that we configured to get only a product category: `PD_ID=34195` for test purposes.

After the MQT creation, a SET INTEGRITY must be executed as shown in Example 8-2:

Example 8-2 Set integrity

```
SET INTEGRITY FOR DWH.AVG_SALES_DAY ALL IMMEDIATE UNCHECKED
```

A refresh was scheduled to execute every night. It is responsible for keeping the most up-to-date information in the MQT. This refresh statement is very simple and is shown in Example 8-3:

Example 8-3 Refresh table

```
REFRESH TABLE DWH.AVG_SALES_DAY
```

The statement used inside our flow to extract the average sales per day is depicted in Example 8-4 on page 292. Because the replication is done from the WebSphere Business Monitor performance warehouse, a similar statement (using the count() function and where the process instance start time range is during the current day) is executed to get the total number of returns for the current day.

Example 8-4 Data warehouse SQL query

```
SELECT AVG(COUNT) AS AVRG_SALES FROM DWH.AVG_SALES_DAY
```

2. DecideAction, Refund, Repair, Exchange, ReceiveCall, CallAnalysis, and SendEmail are only empty Java implementations.

The BPEL diagram for GoodsReturnProcess is depicted in Figure 8-24.

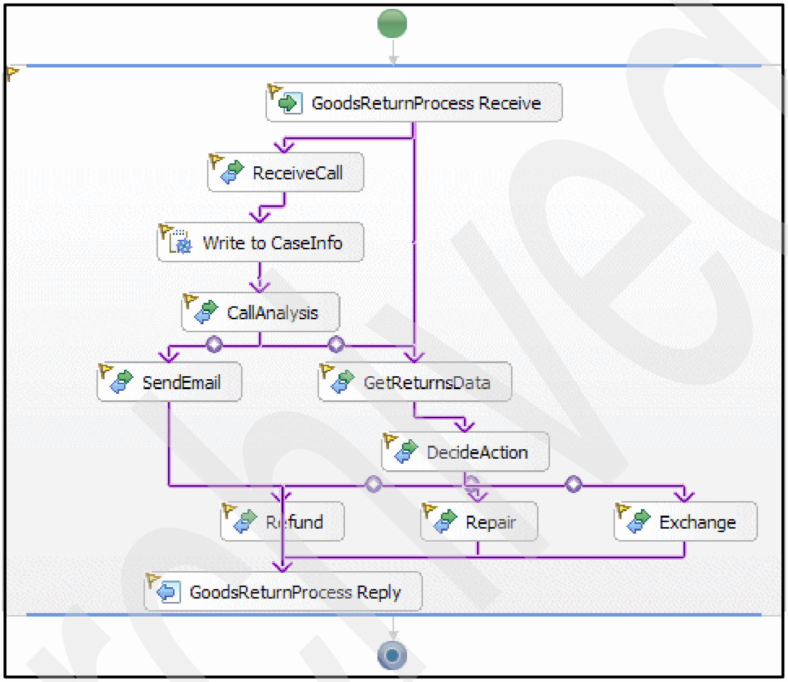


Figure 8-24 GoodsReturnProcess BPEL diagram

Figure 8-25 on page 293 shows the six components of the GoodsReturnProcess.

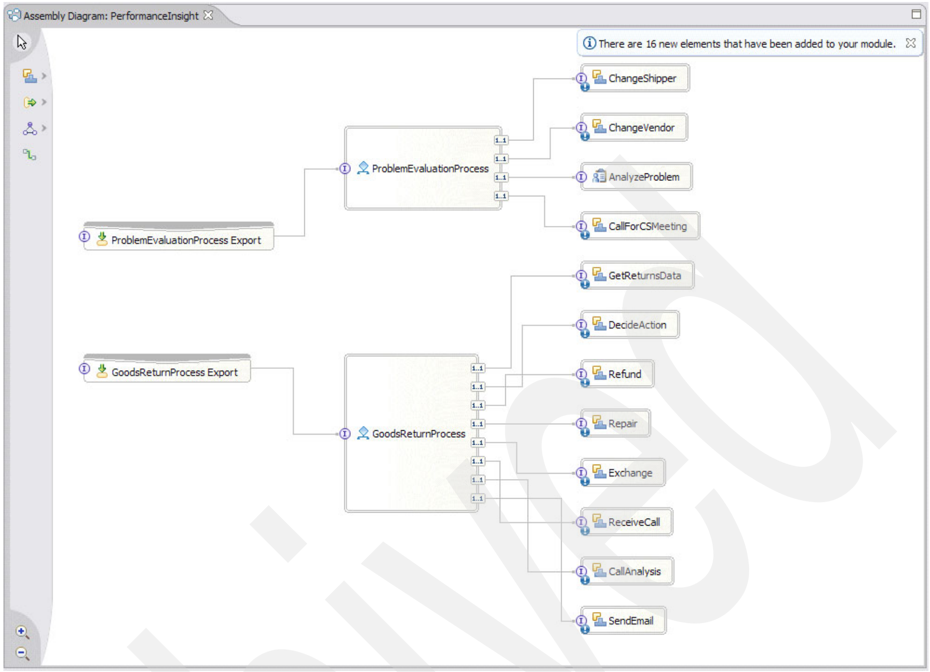


Figure 8-25 PerformanceInsight assembly diagram

Implementing the ProblemEvaluationProcess

The ProblemEvaluationProcess has four activities (see Figure 8-25 and Figure 8-26 on page 294). All of them are implemented as empty Java code except the AnalyzeProblem component, which is realized as a human task.

When you set the implementation type as a Human task in WebSphere Business Modeler, some of the required configurations are generated. However, you still need additional configurations.

To edit in the Human task properties:

1. Double-click the **AnalyzeProblem** component in the assembly diagram to open the implementation. Another way to do this is to double-click the human task **AnalyzeProblem_XXXXXXX** under the **Human Tasks** node in the PerformanceInsight module.
2. The Human Task editor opens showing the configuration of this human task, as depicted in Figure 8-27 on page 295.
3. Click the **AnalyzeProblem_XXXXX** tab at the top of the editor.
4. In the **Properties** view, go to the **Details** tab.
5. Here, we need to specify the user registry that holds the employees information, and who the employees are that are qualified to claim such an activity. However, we choose to work on the Staff Plug-in provided with Process Server, which is only intended for testing purposes.
6. In the **JNDI name of staff plug-in configuration,** choose **bpe/staff/everybodyconfiguration**. This means that anybody can claim this Human task and start working on it.
7. Save your files.

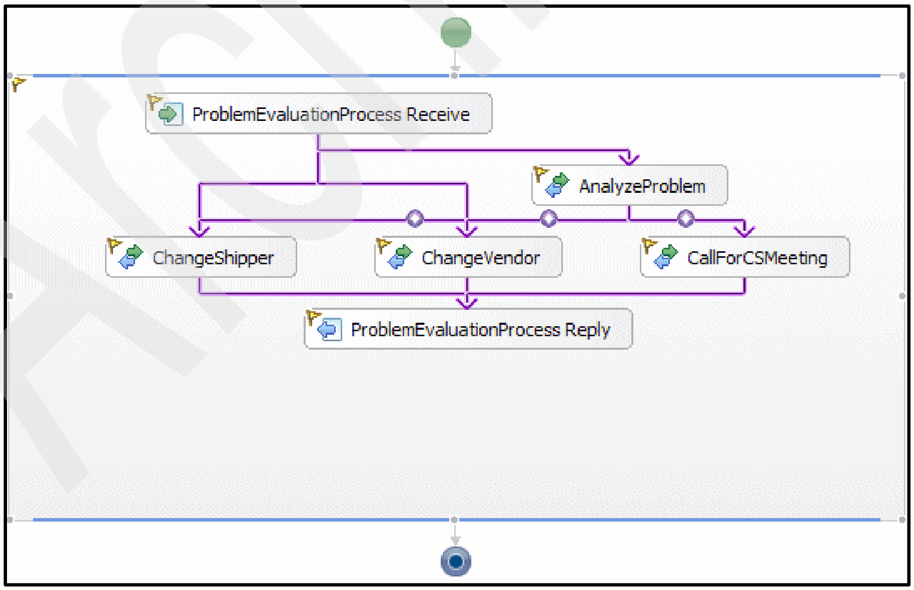


Figure 8-26 ProblemEvaluationProcess BPEL diagram

Figure 8-27 depicts the Human Task editor opening to show the configuration of the Human task.

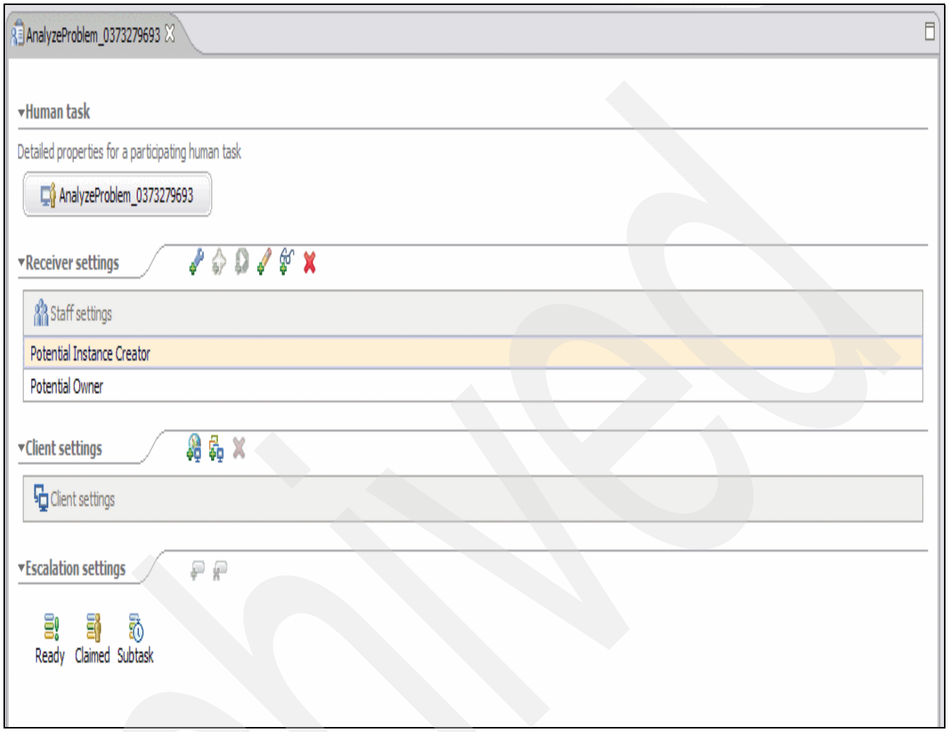


Figure 8-27 Human task editor for AnalyzeProblem

Building the transform mediator module

Mediator is a service component whose sole function is to provide a mediation function between different sets of modules, such as:

- Transforming a message from one format to another so that the receiving service can accept the message
- Conditionally routing a message to one or more target services based on the contents of the message
- Augmenting a message by adding data from a particular data source

The Transform Mediator function is to parse the CBE XML string passed by Adaptive Action Manager and extract two values: *AlarmValue* and *ProductCategory.*

Mediation modules can be realized in two types:

- Mediation flows: Here you define the parameters and functions mapping in a graphical editor.
- Java: Here the implementation is realized as Java code.

Because we are going to parse an XML, we have to use the Java implementation type.

To create a new mediation module:

1. From the **File** menu, click **New** → **Other.**
2. Select **Business Integration** → **Mediation Module**.
3. Click **Next**.
4. Enter the module name `Transformer`.

> **Note:** Try to make your module name as short as possible. Later, we will generate a WSDL interface for the Web service, and the module name will be part of the WSDL file name. If the file path is longer than 200 characters, you will not be able to deploy this module in WebSphere Process Server.

5. Select **WebSphere Process Server V6.0** as the target runtime.
6. Uncheck **Create mediation flow component**. We are going to make a Java component, not a flow component.
7. Click **Next**.
8. Select **PerformanceInsightLib** as the library for the module. We use the interfaces in this library later in the process.
9. Click **Finish**.

Create a new Interface for the Mediator:

1. In the Business Integration view, right-click the **Interfaces** node.
2. Click **New** → **Interface**.
3. Enter the name of the interface `Transformer`. Click **Finish**.
4. The Interface Editor is open, add a request/response operation called `transform` by clicking the **Add Request Response Operation** button at the top. See Figure 8-28 on page 297.
5. Add an Input called `cbe` of type `String`.
6. Add an output called `ret` of type `Int`.

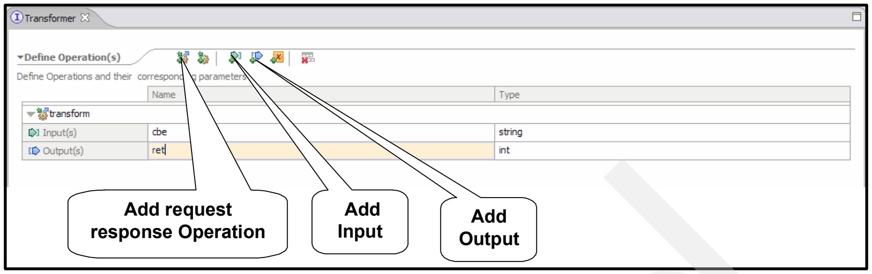


Figure 8-28 Interface Editor

To add a Java component to the Mediator, perform the following:

1. In the Business Integration view, under the Transformer module, open the Transformer assembly diagram by double-clicking the **Transformer** node.
2. From the palette on the left, click the arrow beside the **Mediation Flow** button, and click the **Java** component button.
3. In the Properties view, in the Description tab, type the name of the module, which is `Transformer`.
4. Hold the mouse over the component in the Assembly Diagram, and click the **Add Interface** button.
5. Select the `Transformer` interface.
6. Click **OK**.

Figure 8-29 on page 298 shows the process of adding a Java component to the Mediator.

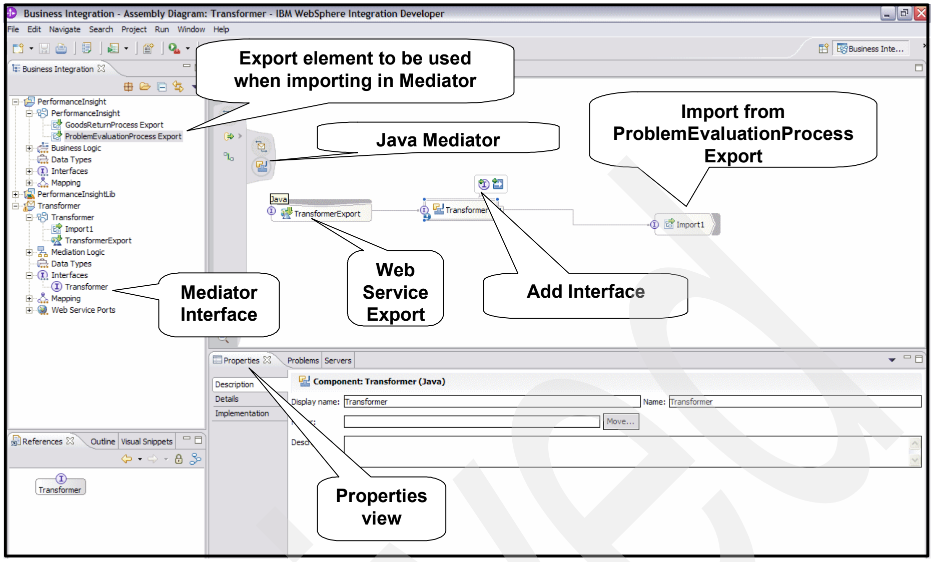


Figure 8-29 Building the mediator

Add the Import to ProblemEvaluationProcess. See Figure 8-29:

1. From the PerformanceInsight module in the Business Integration view, drag the **ProblemEvaluationProcess Export** node and drop it in the Transformer Assembly Diagram.
2. Select **Import with SCA binding**, and click **OK.**
3. Extend a wire from the Transformer module to the new Import element.
4. A pop-up message appears that says: `A matching reference will be created on the source node. Do you want to continue?` Click **OK**.
5. Another message appears asking whether you would like to convert the interfaces from WSDL to Java. Click **Yes.**

Add the Web Service export. See Figure 8-29:

1. Right click the **Transformer** component in the Assembly Diagram, and choose **Export** → **Web Service Binding**.
2. A pop-up message appears asking whether you would like to have the binding/service/port elements defined. Click **Yes**.
3. Select **soap/http** and click **OK**.

Implement the Transformer component:

1. Double-click the **Transformer** component in the assembly diagram.
2. Click **Yes**.
3. Select the package in which you want to add the implementation.
4. The Java editor opens with the implementation class.

The function of the code we are adding is to parse the given CBE XML string and extract two values, AlarmValue and ProductCategory. We then invoke the ProblemEvaluationProcess module asynchronously. Example 8-5 shows a sample CBE XML string. The name of the situation you have defined in WebSphere Business Modeler *HighReturnsRatio* and the date elements *AlarmValue* and *ProductCategory* are highlighted.

Example 8-5 Sample CBE XML string

```
<CommonBaseEvent creationTime="2006-03-13T18:11:57.406Z"
extensionName="ReturnsAlarmtEvent"
globalInstanceId="CE11DAB2BCE9D04FF0C5D2B7707417B4DC" elapsedTime="12000"
priority="50" sequenceNumber="4" severity="10" version="1.0.1">
   <contextDataElements name="ContextID" type="string">
      <contextValue>9501</contextValue>
   </contextDataElements>
   <contextDataElements name="ContextDef" type="string">
      <contextValue>S6DWNUW65L45BYX4ZW645ODHVU</contextValue>
   </contextDataElements>
   <extendedDataElements name="BusinessSituationName" type="string">
      <values>HighReturnsRatio</values>
   </extendedDataElements>
   <extendedDataElements name="AlarmValue" type="double">
      <values>15.0</values>
   </extendedDataElements>
   <extendedDataElements name="ProductCategory" type="string">
      <values>STORAGE</values>
   </extendedDataElements>
   <sourceComponentId application="WebSphere Business Monitor Version 6.0"
component="com.ibm.wbimonitor.observationmgr" componentIdType="ProductName"
location="9.43.86.103" locationType="IPV4" subComponent="com.ibm.wbimonitor"
componentType="Engine"/>
   <situation categoryName="ReportSituation">
      <situationType xsi:type="ReportSituation" reasoningScope="EXTERNAL"
reportCategory="ecode"/>
   </situation>
</CommonBaseEvent>
```

A sample of the code generated is shown in Example 8-6 on page 300. The code simply searches for elements with the name *extendedDataElement*, and checks if their *name* attribute equals *AlarmValue* or *ProductCategory*, extracts their values, and invokes the ProblemEvaluationProcess.

Example 8-6 CBE XML parsing sample code

```
//Add this private function in the class, the function simply compares the
//value in the “name” attribute with the passed name. If the values match,
//it extracts the text in the child element “value”, and returns it.
private String getValue(Element dataElement, String requiredElementName){
   if(dataElement.getAttribute("name").equals(requiredElementName)){
      NodeList values = dataElement.getElementsByTagName("values");
      if(values.getLength() > 0){
         Node value = values.item(0);
         if(value.getChildNodes().getLength() > 0){
            return value.getChildNodes().item(0).getNodeValue();
         }
      }
   }

   return null;
}

//Add this implementation to the tramsform function
public Integer transform(String cbe) {
   try {
      //Create an input source with the given String
      InputSource is = new InputSource(new StringReader(cbe));
      DocumentBuilderFactory factory =
            DocumentBuilderFactory.newInstance();

      DocumentBuilder builder = factory.newDocumentBuilder();
      Document document = builder.parse(is);

      String productCategory = null;
      String alarmValue = null;


      NodeList dataElements =
         document.getElementsByTagName("extendedDataElements");
      //Extracts the ProductCategory value
      for (int i = 0; i < dataElements.getLength(); i++) {
         productCategory = getValue((Element)dataElements.item(i),
                              "ProductCategory");

         if(productCategory != null){
            break;
         }
      }
      //extracts the AlarmValue value
```

```
            for (int i = 0; i < dataElements.getLength(); i++) {
               alarmValue = getValue((Element)dataElements.item(i),
                                 "AlarmValue");

               if(alarmValue != null){
                  break;
               }
            }

            //Invoke the ProblemEvaluationProcess
            ((ProblemEvaluationProcessAsync)
                  locateService_ProblemEvaluationProcessPartner()).
                     InputCriterionAsync(
                           productCategory,
                           Double.parseDouble(alarmValue));

         } catch (Exception e) {
            e.printStackTrace();
         }

         return new Integer(0);
      }
```

8.3.2 Deployment in WebSphere Process Server

To deploy the modules in WebSphere Process Server, you have first to export them from WebSphere Integration Developer. To export the PerformanceInsight and Transformer modules:

1. Click **File** → **Export.**
2. Select **Integration Module**, and click **Next**.
3. Check the box beside the **PerformanceInsight** module. Check the box beside the **Transformer** module.
4. At the bottom, select **EAR file for server deploymen**t.
5. Click **Next**.
6. Browse to the target directory where you want the EAR files saved.
7. Click **Finish**.

To deploy the EAR files in WebSphere Process Server, install them as regular applications from the **Application** → **Install New Application** pane in the DWE Admin Console.

8.3.3 Importing in WebSphere Business Monitor

In this section, we describe how to import the Monitor.zip file (previously exported from WebSphere Business Modeler) into WebSphere Business Monitor to be able to monitor the defined metrics. The steps to import the model in WebSphere Business Monitor are:

1. Generate the database artifacts.
2. Run the database configuration DDLs.
3. Import model into WebSphere Business Monitor Server.
4. Set up the replication.

Generate the database artifacts

In the server, where WebSphere Business Monitor is installed:

1. Go to the administration console of WebSphere Process Server.
2. From the left pane, go to **WebSphere Business Monitor** → **Schema Generator** → **Configuration**.
3. In the **General Configuration** Tab, perform the following:
    a. In the **Table Space Properties File** text box, type the path of the properties file created by the launchpad. Usually, the file path is: **Monitor_Install_Directory\install\mondb\default_minimum_tablespace.properties**
    b. In the **Business Measures Model** text box, type the path of Monitor.zip exported from WebSphere Business Modeler. Note that you have to copy the file first to the WebSphere Business Monitor Server.
    c. In the **Output Directory** text box, type the directory where you want to save the generated files.
    d. If this is the first time you have imported this model (not a new version of the model), check the box to the left of **Ignore older deployments and generate all artifacts**.
    e. Click **OK,** and then click **Save**.

The general configuration tab is depicted in Figure 8-30 on page 303.

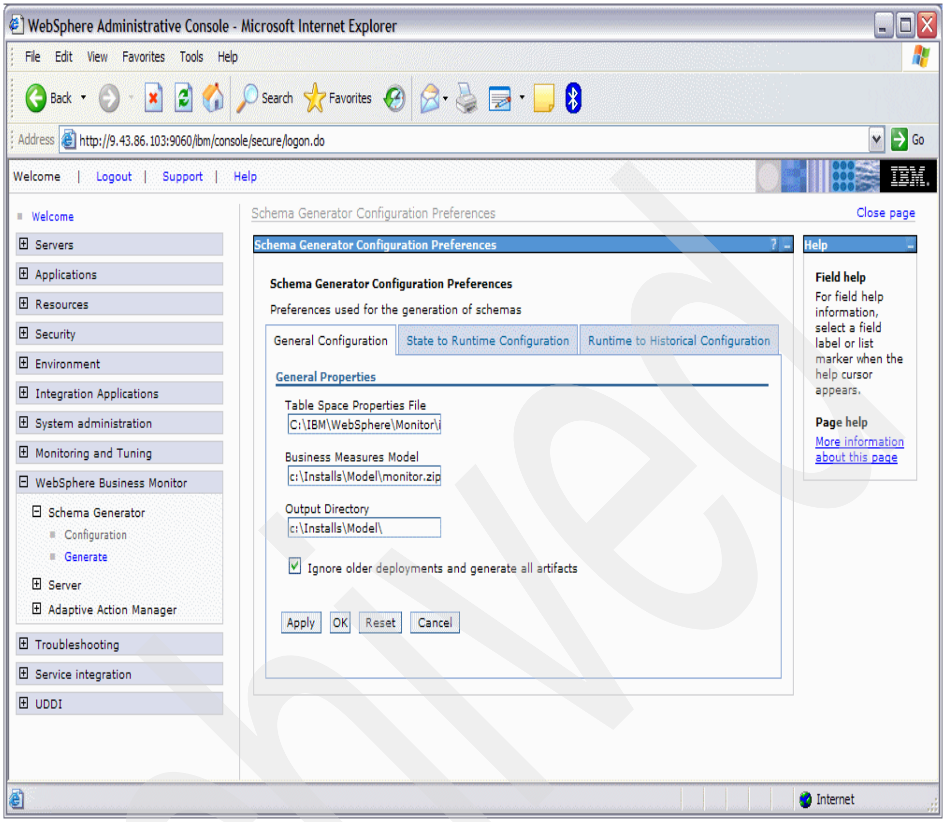


Figure 8-30 General configuration in Schema Generator

4. The State to Runtime Configuration tab lets you configure how the replication will take place from the State database to the Runtime database:

   a. In the **Capture Log Path** text box, type the path where Schema Generator writes the capture replication logs.

   b. In the **Apply Log Path** text box, type the path where Schema Generator writes the apply replication logs.

   c. In **Runtime Database Population Interval**, enter the interval for the replication in minutes. It should be a small interval (about 5 minutes) in order to get real-time information in the dashboards. Note that any instances stored in the State database will not be replicated in the Runtime database until the replication is launched each interval of time.

   d. Click **OK,** and then click **Save.**

5. The Runtime to Historical Configuration tab lets you configure how the replication will take place from the Runtime database to the Historical database. Repeat the same process defined in step 4.

> **Note:** Because our alert is hooked to the ReturnsRatio metric, which is being queried from the data warehouse and which in turn pulls information from WebSphere Business Monitor Historical database through replication, you should make the Runtime-Historical replication interval short in order to trigger the alert in almost real time.

6. To generate the artifacts, go to the **WebSphere Business Monitor** → **Schema Generator** → **Generate** pane.
7. Click **Generate**.
8. A message will appear telling you whether the operation was performed successfully or whether it failed.
9. The artifacts are generated in the output directory you entered before under /schemagen.

Run the database configuration DDLs

In the server where the databases are installed (the WebSphere Business Monitor server in our case), go to the directory where the database artifacts have been generated.

1. Open the DB2 Command Window.
2. Connect to the State database.
3. Run the DDL file state.ddl using the command:

```
db2 -td; -f state.ddl
```

4. Disconnect from the database.

Repeat steps 1-4 for both the Runtime database and the Historical database. The DDL configuration file for the Runtime database is runtime.ddl. And, the DDL configuration file for the Historical database is datamart.ddl.

The generated files are shown in Figure 8-31 on page 305.

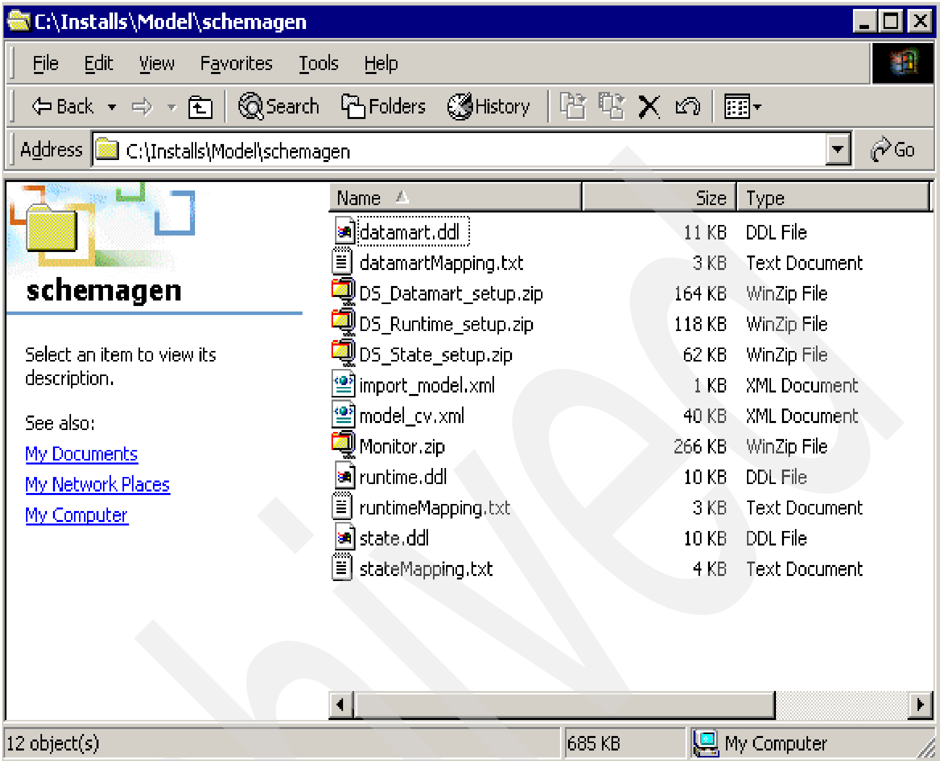


Figure 8-31 Generated DB artifacts

Import model in WebSphere Business Monitor Server

Use the following steps to import the model.

1. In Process Server Admin Console, go to **WebSphere Business Monitor** → **Server** → **Business Measures Model** → **Model Import.**
2. In the **File Name** field, browse to, or type, the path of the Monitor.zip file exported from WebSphere Business Modeler.
3. Click **Import**. The process might take a few minutes to complete.
4. A message will appear telling you whether the operation has been performed successfully or has failed.

Figure 8-32 on page 306 is an example of the import results.

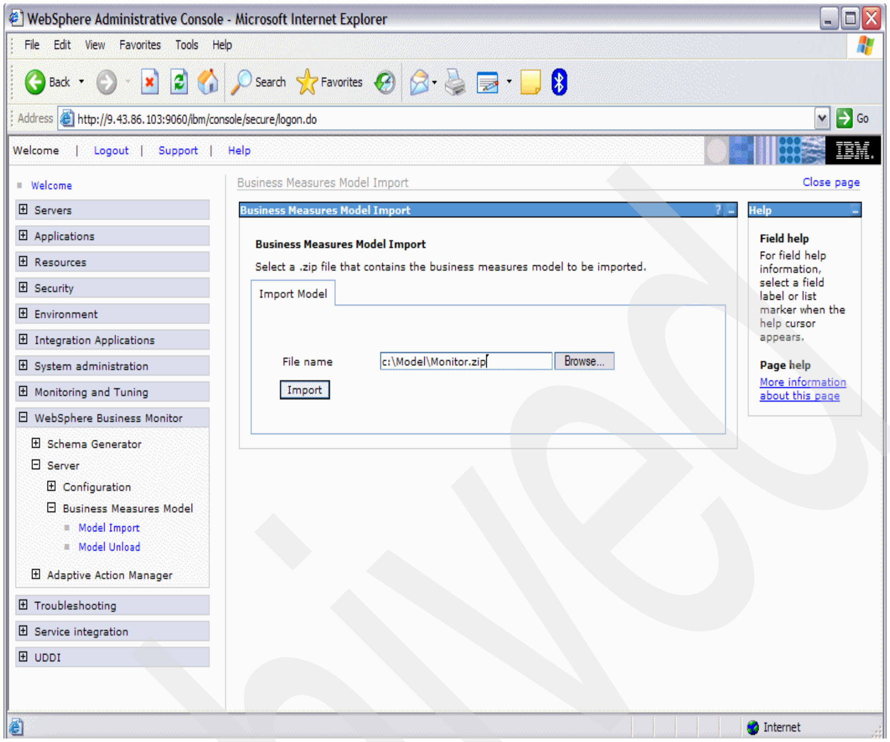


Figure 8-32 Import in WebSphere Business Monitor Server

Set up the replication

Replication takes place from the State database to the Runtime database and from the Runtime database to the Historical database. The following instructions apply only to our particular topology, where all the databases are on the same server. If your topology has the repository and State database on one server, while the Runtime database and the Historical database are on the dashboards server, you will have to make slight changes in the instructions. Refer to WebSphere Business Monitor Information Center for the detailed instructions for your particular topology.

1. In the generated database artifacts directory, you will find three ZIP files:

   a. DS_State_setup.zip
   b. DS_Runtime_setup.zip
   c. DS_Datamart_setup.zip

   Extract them one by one in any order into one target directory, and use the overwrite option.

Note: Make the path of this directory short to avoid potential problems with replication.

2. Open the DB2 Command Window.
3. Go to the directory where you have extracted the three ZIP files. You will find four batch files:
   a. State_to_Runtime_setup_source.bat
   b. State_to_Runtime_setup_target.bat
   c. Runtime_to_Historical_setup_source.bat
   d. Runtime_to_Historical_setup_target.bat
4. Run State_to_Runtime_setup_source.bat, enter DB2 username and password when prompted.
5. At the end, a message will appear telling you whether the operation has been performed successfully or not.
6. Run State_to_Runtime_setup_target.bat, enter DB2 username and password when prompted.
7. At the end, a message will appear telling you whether the operation has been performed successfully or not.
8. Run Runtime_to_Historical_setup_source.bat, enter DB2 username and password when prompted.
9. At the end, a message will appear telling you whether the operation has been performed successfully or not.
10. Run Runtime_to_Historical_setup_target.bat, enter DB2 username and password when prompted.
11. At the end, a message will appear telling you whether the operation has been performed successfully or not.
12. Now run the daemons that perform the replication. These scripts always must be running. So, if the server has been restarted, these scripts have to be rerun. In the directory where you have extracted the ZIP files, you will find two directories:
    a. State_To_Runtime
    b. Runtime_To_Historical
13. Open the DB2 Command Window.
14. In State_To_Runtime/source, run all the files that start by StartCapture*.bat, one by one from the DB2 Command Window. A separate window will be opened for each batch file.

15. In State_To_Runtime/target, run all the files that start by StartApply*.bat, one by one from the DB2 Command Window. A separate window will be opened for each batch file.
16. In Runtime_To_Historical/source, run all the files that start by StartCapture*.bat, one by one from the DB2 Command Window. A separate window will be opened for each batch file.
17. In Runtime_To_Historical/target, run all the files that start by StartApply*.bat, one by one from the DB2 Command Window. A separate window will be opened for each batch file.

8.3.4 Adaptive Action Manager configuration

To automate the process when the returns ratio exceeds the threshold, we create a new instance of the ProblemEvaluationProcess, that will create a new human task for the Product Category Manager to troubleshoot the subject problem.

To do this, we configure the action manager to invoke the Transformer mediator Web services passing the entire CBE XML, once the situation event has been thrown by WebSphere Business Monitor Server. The transformer in turn parses the CBE XML and invokes the ProblemEvaluationProcess.

Defining a Web Service template in Adaptive Action Manager

In Action Manager, you define your required action configuration in a *Template.* Then, you bind the specific type of situation event to one or more action templates. These templates will be executed once the situation event has been thrown. To define a new Web service template, refer to the template in Figure 8-33 on page 312:

1. In the server where WebSphere Business Monitor is installed, in the Process Server Admin Console, go to **WebSphere Business Monitor** → **Adaptive Action Manager** → **Template Definition** → **Web Service** in the left pane.
2. Click **New**.
3. These Web service configurations have to be completed according to the WSDL file created when creating the Web Service Export element for the Transformer module. In WebSphere Integration Developer, switch to the **Resource** perspective.
4. In Transformer folder, right-click: **TransformerExport_TransformerHttp_Service.wsdl**. Select **Open With** → **XML Source Page Editor.**

   This is the WSDL file for the Web service, which imports another WSDL file, Transformer.wsdl, which is the WSDL definition for the Transformer interface we created in the Transformer mediator module. Refer to Example 8-7 and Example 8-8.

Example 8-7 TransformerExport_TransformerHttp_Service.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="TransformerExport_TransformerHttp_Service"
   targetNamespace="http://Transformer/Transformer/Binding"
   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
   xmlns:Port_0="http://Transformer/Transformer"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns:this="http://Transformer/Transformer/Binding"
   xmlns="http://schemas.xmlsoap.org/wsdl/">
   <wsdl:import namespace="http://Transformer/Transformer"
      location="Transformer.wsdl" />
   <wsdl:binding name="TransformerExport_TransformerHttpBinding"
      type="Port_0:Transformer">
      <soap:binding style="document"
         transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="transform">
         <soap:operation />
         <wsdl:input name="transformRequest">
            <soap:body use="literal" />
         </wsdl:input>
         <wsdl:output name="transformResponse">
            <soap:body use="literal" />
         </wsdl:output>
      </wsdl:operation>
   </wsdl:binding>
   <wsdl:service name="TransformerExport_TransformerHttpService">
      <wsdl:port name="TransformerExport_TransformerHttpPort"
         binding="this:TransformerExport_TransformerHttpBinding">
         <soap:address
location="http://localhost:9080/TransformerWeb/sca/TransformerExport" />
      </wsdl:port>
   </wsdl:service>
</wsdl:definitions>
```

The Transformer.wsdl file is the definition of the Transformer interface and is imported by the Web service. It is shown in Example 8-8.

Example 8-8 Transformer.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:tns="http://Transformer/Transformer"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="Transformer"
```

```
   targetNamespace="http://Transformer/Transformer">
   <wsdl:types>
      <xsd:schema targetNamespace="http://Transformer/Transformer"
         xmlns:tns="http://Transformer/Transformer"
         xmlns:xsd="http://www.w3.org/2001/XMLSchema">
         <xsd:element name="transform">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="cbe" nillable="true"
                     type="xsd:string" />
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
         <xsd:element name="transformResponse">
            <xsd:complexType>
               <xsd:sequence>
                  <xsd:element name="return" nillable="true"
                     type="xsd:int" />
               </xsd:sequence>
            </xsd:complexType>
         </xsd:element>
      </xsd:schema>
   </wsdl:types>
   <wsdl:message name="transformRequestMsg">
      <wsdl:part element="tns:transform" name="transformParameters" />
   </wsdl:message>
   <wsdl:message name="transformResponseMsg">
      <wsdl:part element="tns:transformResponse"
         name="transformResult" />
   </wsdl:message>
   <wsdl:portType name="Transformer">
      <wsdl:operation name="transform">
         <wsdl:input message="tns:transformRequestMsg"
            name="transformRequest" />
         <wsdl:output message="tns:transformResponseMsg"
            name="transformResponse" />
      </wsdl:operation>
   </wsdl:portType>
</wsdl:definitions>
```

5. According to the WSDL files, fill in the form as follows:
   a. **Template name**: Type any name, for example, `SituationMediation`.
   b. **Description**: Type any description for the service, or leave it blank.
   c. **Target Namespace**: Type `http://Transformer/Transformer`

      In Example 8-7 on page 309 and Example 8-8 on page 309, notice that the service definition itself is defined in namespace http://Transformer/Transformer/Binding while other parts that are defined in Transfomer.wsdl and imported in TransformerExport_TransformerHttp_Service.wsdl are under the namespace http://Transformer/Transformer. Because Action Manager accepts only one namespace in which everything is defined, we enter here `http://Transformer/Transformer`, and make the other names relative to this namespace as we will see when we enter the service name.

   d. **Service name**: Type:
   `/Binding/TransformerExport_TransformerHttpService`

      From Example 8-7 on page 309, note that the service definition is under namespace http://Transformer/Transformer/Binding and the service name is TransformerExport_TransformerHttpService. Because we are using http://Transformer/Transformer as the name space, we enter the service name relative to this namspace, which is:
   `/Binding/TransformerExport_TransformerHttpService`

   e. **End Point address**: Type:
   `http://server_name:9080/TransformerWeb/sca/TransformerExport`

      This is the value of the location attribute under the address element. Change it from localhost to the server address where the Transformer Mediator has been deployed.

   f. **Port type**: Type: `TransformerExport_TransformerHttpPort`

      This is the name of the port we are using.

   g. **Operation name**: `transform`

   h. **Input message name**: `cbe`

      This is the name of the parameter sent in the request message.

6. Click **OK**, and then click **Save**.

The template configuration is depicted in Figure 8-33 on page 312.

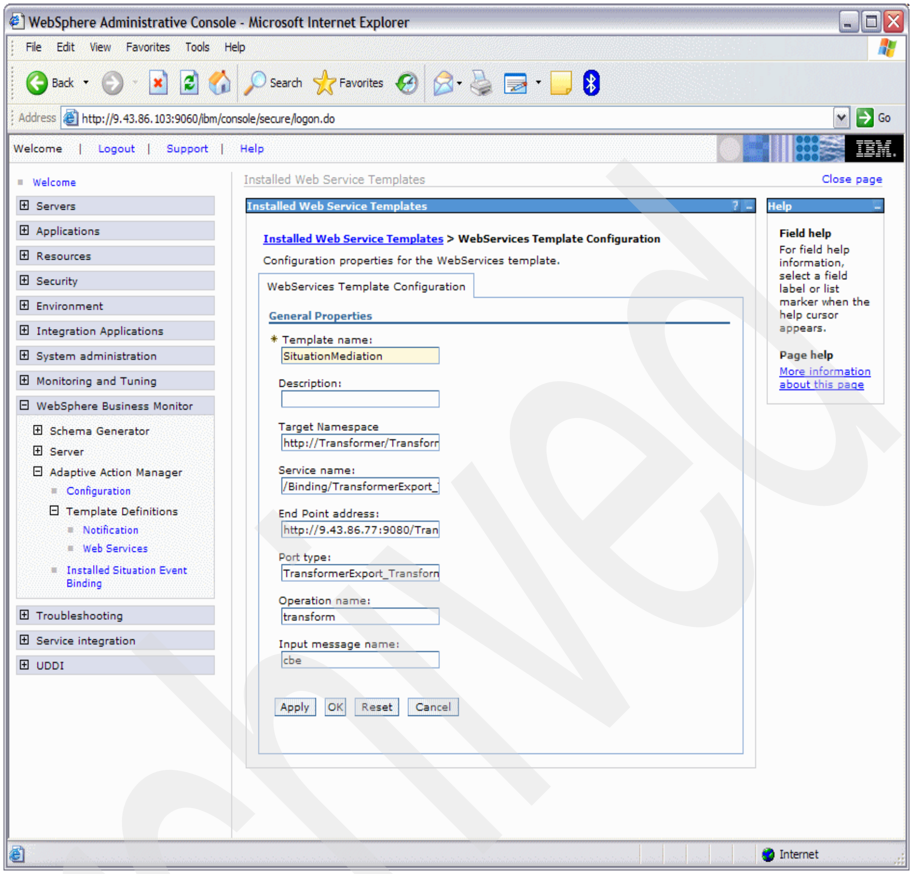


Figure 8-33 Web Service template configuration

Define the binding between the template and the situation

Here you have to define the action template to be executed when a particular situation is detected.

1. Go to **WebSphere Business Monitor** → **Adaptive Action Manager** → **Installed Situation Event Binding**.
2. Click **New**.
3. In **Situation Event Name**, type the name of the business situation you have defined in WebSphere Business Modeler. This is the name the Action Manager uses to bind the event and the template.

Note: This is not the name of the event definition itself. This is the name of the BusinessSituationName you enter when defining the situation event in WebSphere Business Modeler. See Figure 8-34 for an example.

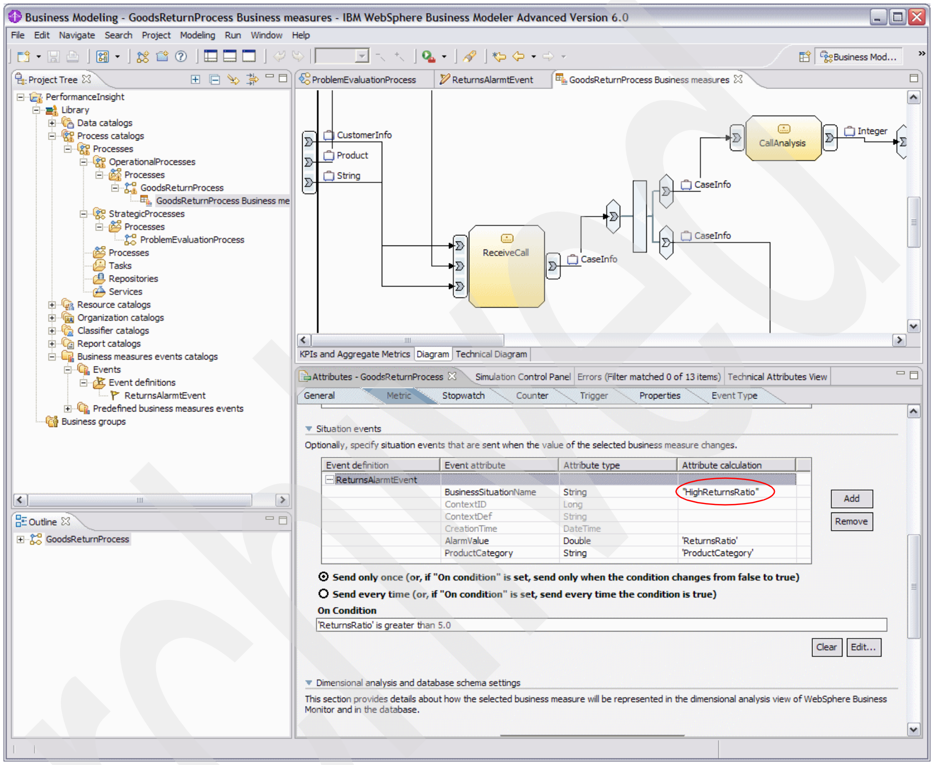


Figure 8-34 BusinessSituationName in WebSphere Business Modeler

4. In our case study, we entered `HighReturnsRatio`. See Figure 8-35 on page 314.
5. Enter a description if needed (it can be blank).
6. Click **OK**.
7. Now click the newly created **HighReturnsRatio** binding to add the templates.
8. On the table at the bottom, click **New**.
9. Select the template you have just created.
10. Click **OK**.

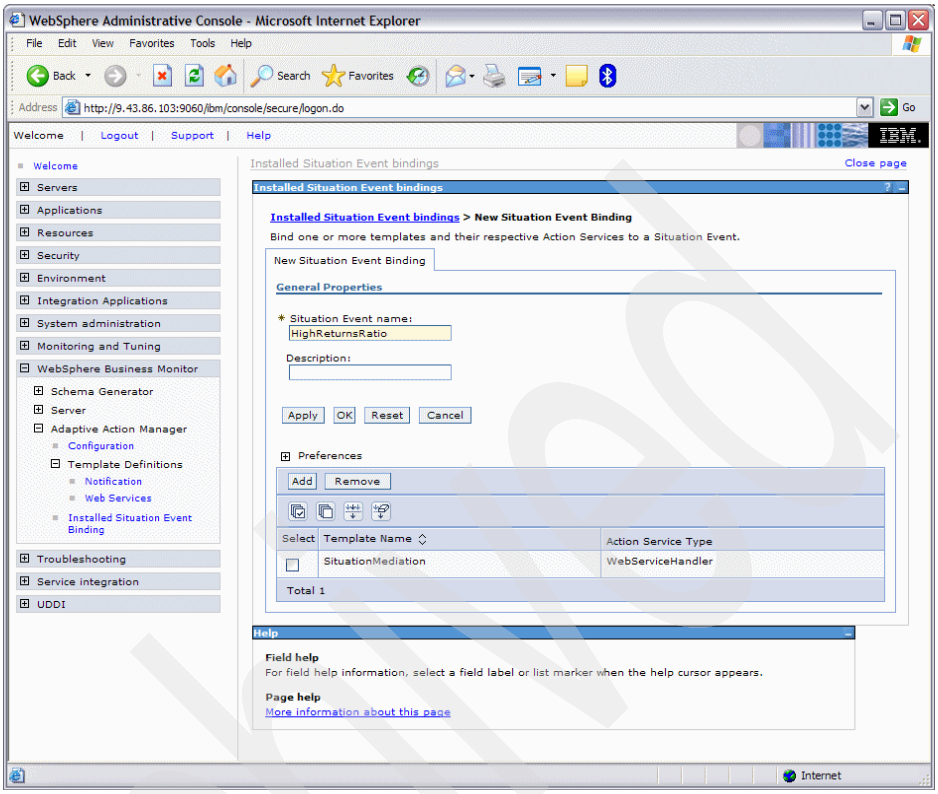


Figure 8-35 Defining situation event binding in Action Manager

Important: You have to restart the Action Manager application for the changes to take effect. From the **Enterprise Applications** pane in Process Server Admin Console, stop and start the application: **IBM_WB_ACTIONMANAGER**.

8.4 Integrating the information

Information Integration is a very valuable technology that is available to support Business Innovation and Optimization. This is because of the critical need to consolidate and integrate information in real time. It has become extremely important for solution development to gain a business advantage and to provide new services for fast and efficient application development and implementation.

The IBM products that play a key role in information integration are described in some detail in Chapter 6, “Case study software components” on page 123.

There are a number approaches for implementing information integration, such as replication, or federation, or a combination of the two. Also, some, or all, of the data could be populated to the data warehouse by operational applications.

In our case study implementation, we defined replication as the primary approach to integrate information from two disparate sources, the WebSphere Business Monitor history database and the enterprise data warehouse. This replication was performed as a regular and typical BI extraction.

The replication could have been done by using Q Replication, Federation, or a regular ETL job. The data transformation is a particular process that varies from project to project, based on what you have implemented in your data warehouse. In most projects, the transformation rules are unique and require consideration of their own particular environment.

In this case study, an SQL Replication process was created to expose one of the metrics in the data warehouse for further analysis. The selected table related to the number of product returns, but it could be any information used in data warehouse data flows.

To replicate the table, we first had to analyze the WebSphere Business Monitor cubes. This analysis was executed with the help of the DWE OLAP product. Based on the dimensions defined in WebSphere Business Monitor, a cube was created over the returns process model. The visualization of this cube in the History database, using OLAP Center, is depicted in Figure 8-36 on page 316.

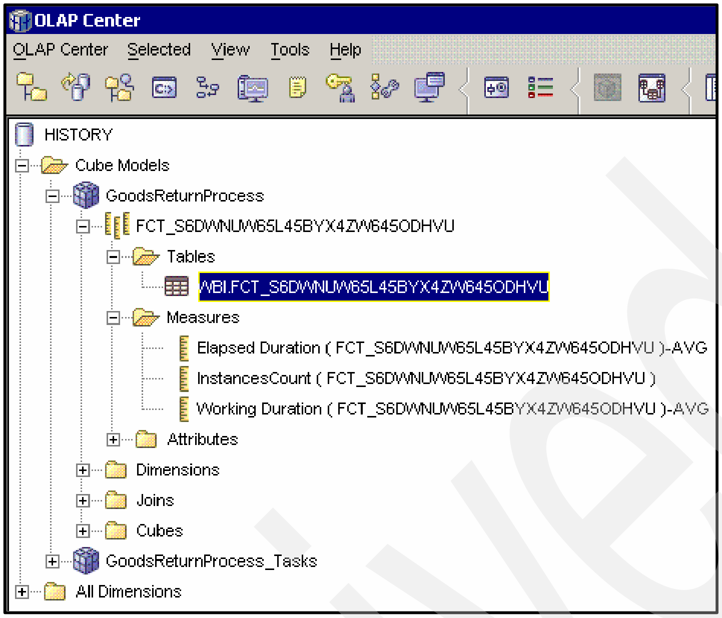


Figure 8-36 Return process cube and measures

The star schema of the Returns process cube is depicted in Figure 8-37 on page 317.

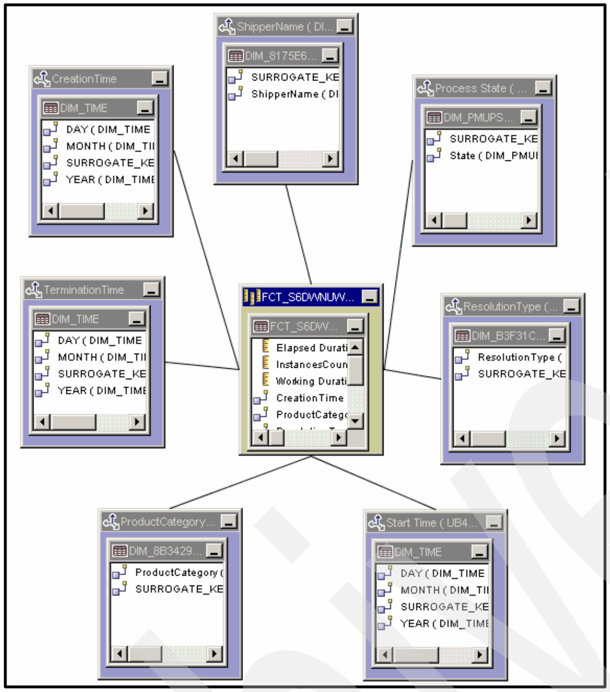


Figure 8-37 Return process schema

Information about the number of returns is acquired by the use of a count over the rows on the fact table that apply to returns due to a specific reason. In our case study, that reason was for a defective product.

In our case study, SQL replication resolved the need. However, in another scenario there might be a requirement for a different approach, perhaps an approach that uses Q replication and federation. Another key capability of information integration is the use of Master Data Management to standardize information over all the sources. This is crucial, for example, to enable applications and databases to speak in the same language. The use of MDM is further described in 5.2.3, “Master Data Management” on page 99.

8.5 Business intelligence modeling

In this scenario, the data warehouse server contains a data mart, included specifically to keep the historical information about product returns. This information provides insights to the Product Category Manager to help identify the root causes of excessive product returns and perform corrective actions.

8.5.1 The relational data model

To provide flexibility in the analysis for the Product Category Manager, we implemented a star schema data model. The model includes the following dimensions and fact table:

- **Product Dimension**: The *product* dimension contains information about each product. Because there are many products in the database, they are categorized in groups based on common characteristics.
- **Return Type Dimension:** The *return type* dimension is a classification dimension and contains information about the reasons that a product is returned by a customer. It includes examples such as damaged product and defective product.
- **Shipper Dimension:** The *shipper* dimension contains information about all shipping services providers.
- **Return Date Dimension:** The *return date* dimension is a time dimension and it contains information about all periods of time in which product returns occurred. To enable the Product Category Manager to perform time series analysis (for example, Year-to-Date Returns, Quarter-to-Date Returns, and Month-to-Date Returns) as well as perform comparisons of different periods (for example, This Years Returns versus Last Years Returns), we organized the hierarchy of this dimension to include the levels Years, Quarters, Months, and Days.
- **Shipping Costs Fact Table:** The *shipping costs* fact tables contains *shipping costs* and *shipping return costs* information for all products by shipper and return type for the last three years.

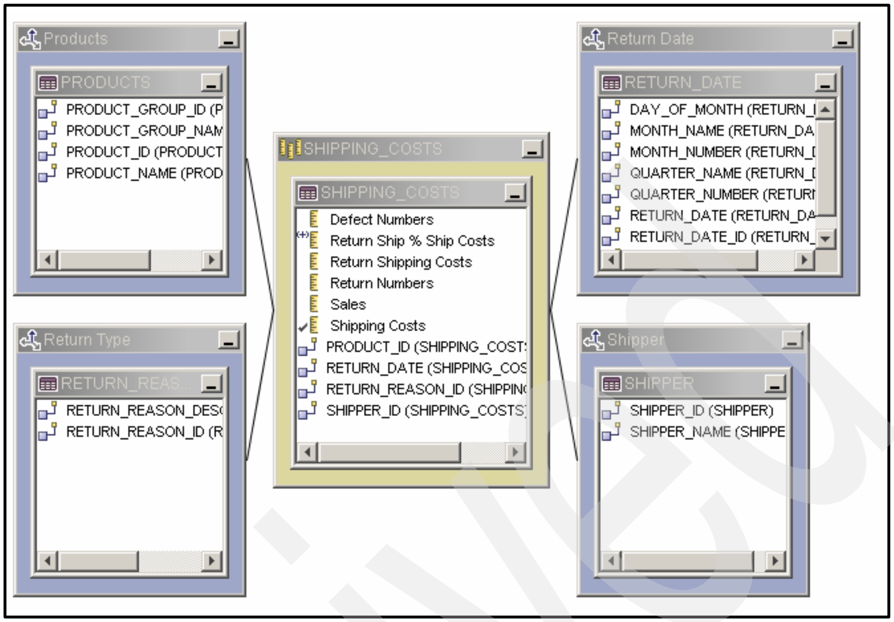


Figure 8-38 Product Returns Data Model

8.5.2 DWE OLAP: The product returns cube model

IBM DWE OLAP is an add-on feature of DB2 UDB that enables DB2 to be OLAP-aware. DWE OLAP is used to create multidimensional cube models on top of a relational DB2 database. Each cube model has its own dimension names and measure names to identify, which are called *object names*. Each object name has an alias to display, which is called *business name*. The cube model metadata is stored on a repository that can be exposed by an XML API to reporting and BI tools, such as DB2 Alphablox. This metadata is also used to optimize the database for faster query performance.

DWE OLAP contains the following components:

- **Catalog Tables**: Used to store the DWE OLAP metadata.
- **OLAP Center GUI**: Windows GUI used to create a cube model and cubes.
- **Cube Model**. A OLAP model that contain measures, dimensions, and join information for a star schema or snow-flake model.
- **Cubes**. Cubes are a subset of the cube model and also contain dimensions and measures.
- **Optimization Advisor**. The Optimization Advisor is a performance advisor that recommends MQTs to improve performance of the queries.

Using the OLAP Center GUI, we created the following cube model:

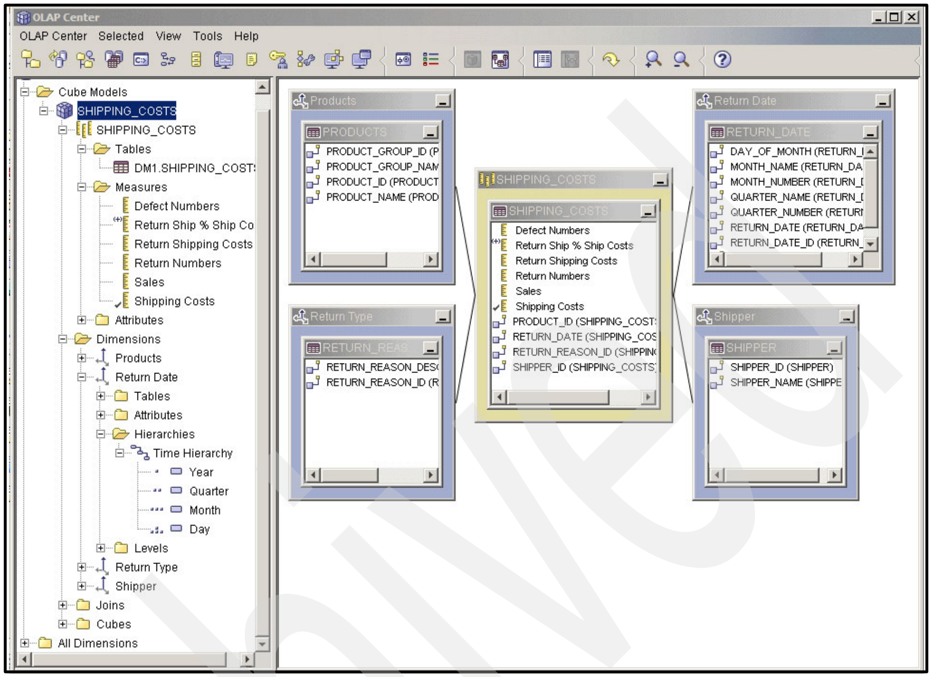


Figure 8-39 Product returns cube model

In the product returns cube model, we defined the following measures:

- Shipping Costs
- Return Shipping Costs
- Return Ship % Ship Costs
- Sales
- Defect Numbers
- Return Numbers

Those measures can be analyzed across the following dimensions and hierarchies:

- Return Date
- Return Type
- Products
- Shipper

The *Return Date* dimension includes a hierarchy allowing analysis at Year, Quarter, Month, and Day levels.

The *Product* dimension has only two levels in the hierarchy, Product Group and Product Number.

The *Shipper* and the *Return Type* dimension hierarchies each contain only one level.

A DWE OLAP cube can be a subset of the cube model (some dimensions and some measures), or it can contain all dimensions and all measures. The DWE OLAP Cube is taken into consideration by the DWE OLAP Optimization Advisor when determining MQT recommendations. We also defined a DWE OLAP cube to feed the metadata to the DB2 Alphablox application. Although this metadata can also be defined in the DB2 Alphablox Administration pages, we recommend that it is defined in the DWE OLAP GUI, because the Cube metadata can be used for performance optimization of the model as well as to build the analytical application.

In our scenario, we did not create an MQT to improve performance of the queries because we were working with a small volume of data. However, we recommend such an approach when dealing with large numbers of concurrent users and large databases.

We created a cube to be used by the DB2 Alphablox analytic application. As you can see in Figure 8-40 on page 322, the *Returns Cube Model* contains all the dimensions and measures that were defined on the *Returns Model* (Figure 8-39 on page 320).

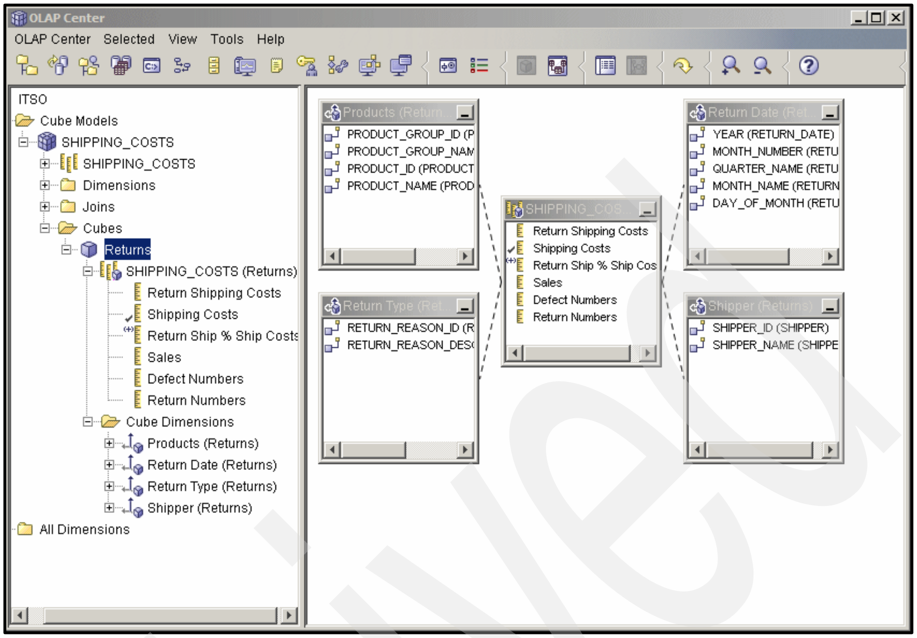


Figure 8-40 Returns Cube model

8.5.3 DB2 Alphablox

DB2 Alphablox is a premier application development platform for rapidly building and broadly deploying custom analytic solutions across the enterprise. DB2 Alphablox provides a set of analytic components and supporting services to make it easy to rapidly assemble analytic applications using JSP™ tags. DB2 Alphablox developers use adapters to connect to all types of data sources and then use data Blox to retrieve data from those data sources and visual Blox to create highly interactive graphs, charts, and reports tailored to suit user needs.

8.5.4 DWE OLAP and Alphablox Integration

In the DB2 OLAP Center, DWE OLAP builds the OLAP Metadata Model, which is transferred from the star schema in DB2 UDB. At the same time with the Optimization Advisor, MQTs can be built in DB2 UDB to improve the query performance. Through the use of the Metadata Bridge, you can import the Cube model into the Cube Manager of DB2 Alphablox and then you can customize it. DB2 Alphablox manages and tunes these imported cubes. Cube Server periodically extracts data from MQTs and fact tables. When the DB2 Alphablox application sends a Multidimensional Expressions (MDX) query, DB2 Alphablox compiles and executes it. Then DB2 Alphablox gets a result set object from the in-memory cache and sends the results back to the Web browser. Figure 8-41 depicts this.

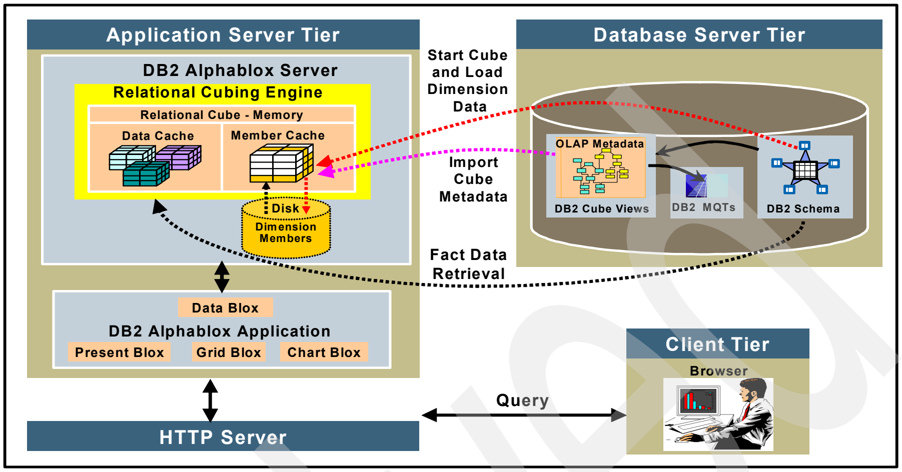


Figure 8-41 DWE OLAP and Alphablox cube manager integration

The integration of DWE OLAP and Alphablox allows creation of a very robust analytical environment to support near real-time analysis. Because the data is physically stored only in the relational database (DB2 in this case), there is no delay to extract and populate the Alphablox Cubes because the data is retrieved on demand (as requested by users).

8.6 Application integration

In this section, we discuss integration issues at the portal level. Dashboards are powerful tools, which can be used for application integration, because they can be used to present a single view of performance insight through a single application. For the implementation of the case study, we used the following software products to build and integrate the performance insight dashboard:

We used these products for infrastructure:

- WebSphere Application Server 6.0.1
- WebSphere Portal 5.1.0.2
- WebSphere Process Server 6.0.1
- DB2 8.2 Fix Pack 11 (DB2 UDB)
- DB2 Alphablox V8.4 (Alphablox)

We used these development tools:

- Rational Application Developer 6.0.1
- WebSphere Integration Developer 6.0.1

For the case study, we created a dashboard to alert the Product Category Manager when a potential for a high number of returns for a product category is detected. This allows the Product Category Manager to analyze the problem, find the root cause, and take an action to resolve the problem.

The dashboard consists of four portlets and is depicted in Figure 8-42.

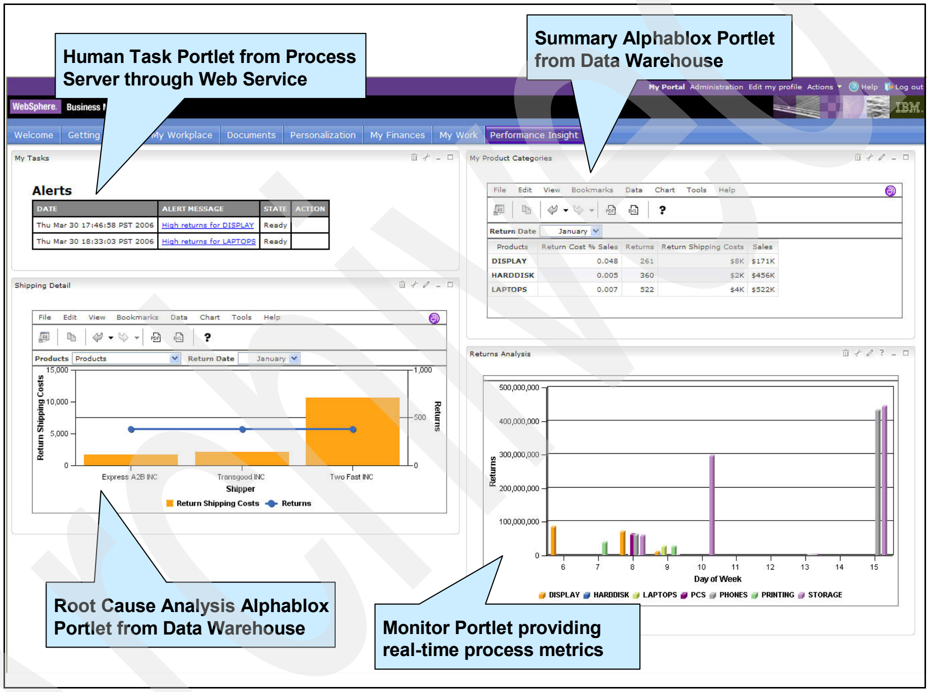


Figure 8-42 Dashboard example

The following list describes the four portlets in Figure 8-42 on page 324:

- My Tasks portlet is a custom portlet that consists of two components. The first component is a list of tasks for the Product Category Manager that are retrieved from the WebSphere Process Server using a Web service interface. When a task gets claimed, a message is sent to the other portlets with the product category code.

  The second component of the portlet allows the Product Category Manager to take the action to change the shipper. This completes the current task and starts a new process to change the shipper. The portlet listens to the messages of the other portlets and will invoke the user interface (UI) to change the shipper.

  For more information, see 8.6.1, “My Tasks portlet” on page 326.

- Analyze portlet is a custom Alphablox portlet built on top of a generic Alphablox portlet that allows the Product Category Manager to analyze return measures across the product hierarchy. The portlet can receive messages from other portlets. In this scenario, the portlet receives product group information from the Alerts and Action Human Task portlet. The portlet can also sends Blox-to-Blox messages, which are invoked by a double-click any data cell, to the Root Cause Analysis portlet about the member names for the current data intersection.
- Process Monitor Analysis portlet allows the Product Category Manager to analyze the metrics for the processes that are currently running in the WebSphere Process Server across a set of dimensions.
- Root Cause Analysis portlet is part of the guided analysis and is based on the same generic Alphablox portlet as the Analyze portlet. In this case, it is configured by the administrator to allow analysis of different shippers for a given time point and product. It accepts Blox-to-Blox communication from the Analyze portlet and slices the data according to the selected data intersection for Time and Product in the Analyze portlet.

Guided Analysis

Analysis of problems and large data sets can be complex. In general, only power users perform such functions, using spreadsheet tools such as Excel or specialized tools. OLAP structures make it easier for users to navigate through the data as the data gets translated into business terms such as measures and dimensions.

Guided Analysis takes this a step further and can enable non-power users to perform such analyses. This opens the problem analysis and solution activity to a wider range of users. Basically, it involves taking the analysis and problem solving knowledge, processes, and capability from experienced successful analysts (or power users) and embedding it into the software. That software can then guide the users through an analysis and problem solving activity, much like an automated process. It represents reuse of skills and experience by a user less knowledgeable and experienced with particular processes and data.

In the case study, we show how a product manager is guided through the analysis of a high number of returns and then determines a root cause analysis of the problem. Note, there could be several potential causes discovered on the way to finding the root cause in a typical customer environment.

Cooperative portlets

There are a number of ways to achieve cooperative portlets. In this case study, we look at two of them.

Cooperative portlets support a loosely-coupled model for inter-portlet communication. Each portlet is associated with a set of typed data items, which are called *properties*, which the portlet can generate or receive. Each portlet can also be associated with a set of actions, which are capable of generating or receiving property values at runtime.

WebSphere Portal provides several ways to achieve cooperative portlets. With Version 5.1, it is only possible using the IBM portlet API and not with the JSR portlets. We expect this will change over time.

8.6.1 My Tasks portlet

The My Tasks portlet returns a list of Human Tasks from the process server as defined in 8.3, "Process implementation and deployment" on page 288 and provides a link that will change the product group on the other portlets through cooperative portlets.

In WebSphere Process Server 6.0.1, there are no client libraries. Therefore, we use a Web service to expose the Process Server API to the WebSphere Portal server.

The My Tasks portlet implementation, described in this section, is depicted in Figure 8-43 on page 327. The portlet will allow users to claim a task and send the product group, which is part of the input message of the task to the Alphablox portlet.

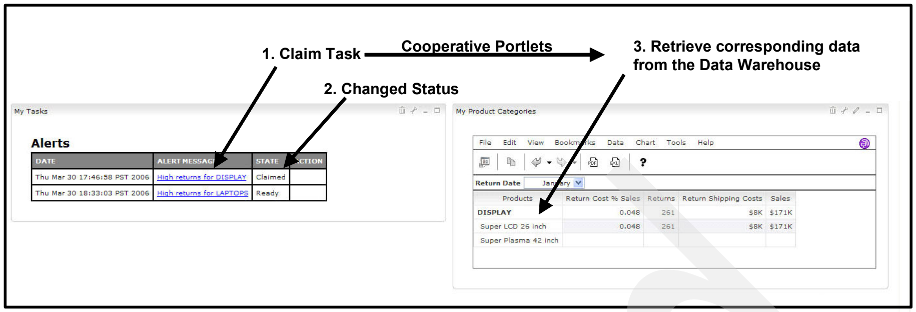


Figure 8-43 My Tasks portlet

The following section is taken in large part from an IBM white paper, *Enabling Generic Web Services Interfaces for Business Process Choreographer,* by Eric Erpenbach and Anh-Khoa Phan.

Review business flow and Human task APIs

It is important to have a general understanding of the capabilities of the BFM and HTM APIs, because we will want to expose certain methods or a combination of methods as services. The APIs cannot be exposed directly as services because of the mapping of the API parameters to the message parts of the interfaces. When considering which APIs to make available as services, it is best to consider which type of business operation we would like to perform and whether we want to combine multiple APIs under a single method to be exposed as a single operation.

For this case study, we chose the following business operations to be exposed as services. This requires a number of HTM API calls. The business operations are:

- Create and start a task
- Claim a task
- Complete a task
- Query a task by ID
- Query for a collection of tasks
- Query for a collection of task IDs
- Get input message for a task ID and property name
- Get output message for a task ID and property name

The JavaDoc for these APIs is available at:

- Human Tasks Manager (HTM)

  {*$WPS_Install_Dir*}\web\apidocs\com\ibm\task\api\

  or

{*$WID_Install_Dir*}\runtimes\bi_v6\web\apidocs\com\ibm\task\api\

- Business Flow Manager (BFM)

  {*$WPS_Install_Dir*}\web\apidocs\com\ibm\bpe\api\

  or

  {*$WID_Install_Dir*}\runtimes\bi_v6\web\apidocs\com\ibm\bpe\api\

Details on using the BFM and HTM APIs are also available in the WebSphere Process Server Information Center at the URL:

`http://www.ibm.com/software/integration/wps/library/infocenter/`

To create and start a task, the task template must be retrieved from the Human Task Manager. This API returns an array of query templates. With the proper "whereClause" value specified, it will return a single template:

LocalHumanTaskManager.queryTaskTemplates((java.lang.String whereClause, java.lang.String orderByClause, java.lang.Integer threshold, java.util.TimeZone timeZone)

With the task template retrieved, the input message for the task can be created in preparation for passing the correct information into the task when it is started. This API will return an empty input message object:

LocalHumanTaskManager.createInputMessage(TKIID tkiid)

With the task obtained and created, a task can be created and started. This API returns a TKIID which is the unique identifier of the task:

LocalHumanTaskManager.createAndStartTask(java.lang.String name, java.lang.String namespace, ClientObjectWrapper input, ReplyHandlerWrapper replyHandler)

Define interface

Once we have chosen the business operations and determined which APIs will be need to be called, we can define the interface. Using WebSphere Integration Developer, create a library project:

File → New → Other → Library

For our example, name the library project: `BFMHTMLibrary`. The library project will be used to hold all interfaces and business objects because this allows for easy sharing inside WebSphere Integration Developer as well as by clients, who will use the interfaces and data objects to call the services. Create the interface as shown in Figure 8-44 on page 329.

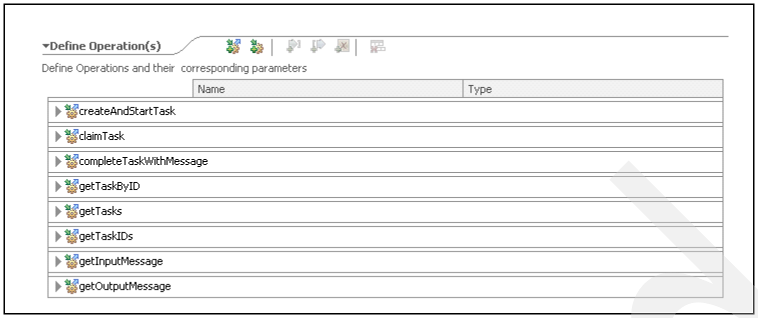


Figure 8-44 Define the HumanTaskManager interface

Before we create the request and response operations, we first have to define some business object definitions.

When retrieving a task by ID or specific string, there are numerous attributes which are available. We can achieve reuse and simplicity in your interface definition by using a business object definition for a task and its attributes. We can see the different attributes of a task which can be returned by examining the predefined Task view in the Information Center:

`http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wsps.javadoc.doc/doc/com/ibm/task/api/Task.html`

A business object definition of the task should include all attributes which we would like to pass back to a client who issued the query request. There are a number of attributes which do not need to be defined in the business object because the attributes are specific to the implementation and will not be logical on a non-WebSphere Process Server environment. Create a business object named `Task` in your library:

File → New → Business Object

Create attributes with the appropriate types for the business object as shown in Figure 8-45 on page 330. This business object will be used in the operation for retrieving a task by specific ID and retrieving a set of tasks based on a specific search string.

Then create another business object, which will be used for the response from the query and which returns a set of tasks based upon a specific query. Currently, the Interface Editor in WebSphere Integration Developer does not allow for an array object to be specified in a message part. In order to work around this, use a business object which includes the array. Create a business object named `QueryResultTask` and add an attribute which is an array of Tasks as shown in Figure 8-45 on page 330.

Last, for the query operation for retrieving a set of IDs, create a business object named `QueryResultTaskID`, which has an array of strings for the IDs as shown in Figure 8-45. Here again, we need to include the array in the business object.

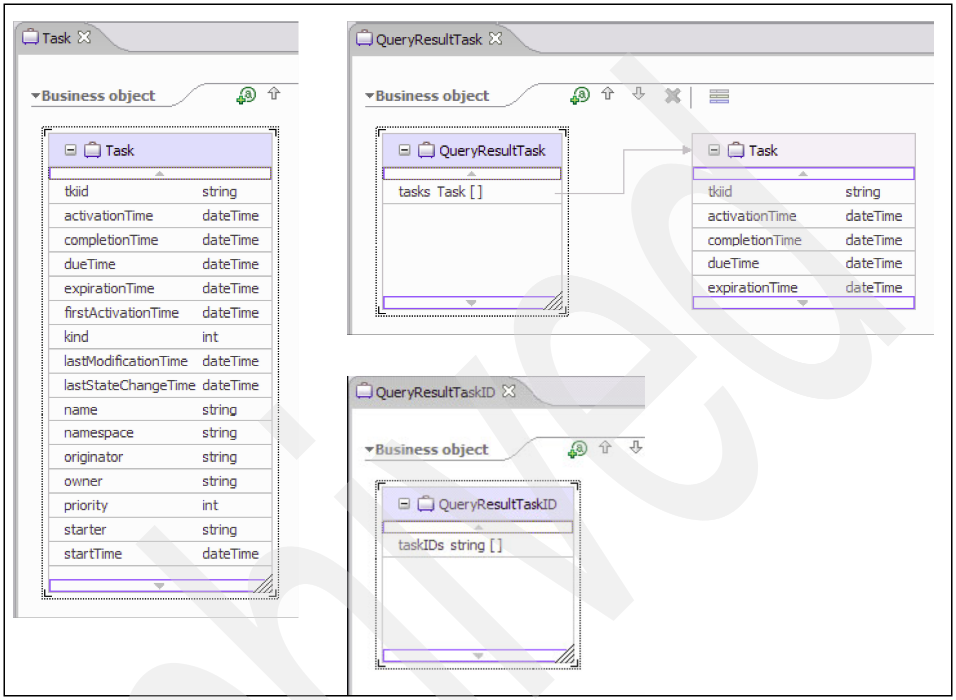


Figure 8-45 Task and QueryResult Business objects

Then create request/response operations for each of the methods, as shown in Table 8-2.

Note: To select a type of `anyType`, click the type field, select **Browse**, then check **Show all XSD types,** and select **anyType**.

Table 8-2 HumanTaskManager Interface Request/Response properties

Method/Type	Property	Type
createAndStartTask		

Method/Type	Property	Type
Input	taskName	string
	taskNamespace	string
	inputMessage	anyType
	replyHandlerWrapper	anyType
Output	tkiid	string
Faults	faultMessage	string
claimTask		
Input	tkiid	string
Output	inputMessage	anyType
Faults	faultMessage	string
completeTaskWithMessage		
Input	tkiid	string
Output	inputMessage	anyType
Faults	faultMessage	string
getTaskByID		
Input	tkiid	string
Output	task	Task
Faults	faultMessage	string
getTasks		
Input	whereClause	string
	orderBy	string
	skipTuples	int
	threshold	int
Output	resultSet	QueryResultTask
Faults	faultMessage	string
getTaskIDs		

Method/Type	Property	Type
Input	whereClause	string
	orderBy	string
	skipTuples	int
	threshold	int
Output	resultSet	QueryResultTaskID
Faults	faultMessage	string
getInputMessage		
Input	tkiid	string
	property	string
Output	inputMessage	string
Faults	faultMessage	string
getOutputMessage		
Input	tkiid	string
	property	string
Output	inputMessage	string
Faults	faultMessage	string

Create implementation and Web service binding

With the interface fully defined, we can create the Java component and call the appropriate HTM APIs to create and start a task. To create the Java component:

1. Create a module called `BFMHTMFacade` by clicking **New** → **Other** → **Module.**
2. Establish a dependency to the library using the Dependency Editor by right-clicking the `BFMHTMFacade` module.
3. Select **Open Dependency Editor**, and then add the **HumanTaskManager** library by clicking **Add**. This will make the interface and business objects available for components in the module.
4. Open the Assembly editor for the module by double-clicking the `BFMHTMFacade` assembly diagram.
5. Drag the interface from the library and drop it on the Assembly editor. Select **Component with No Implementation Type**. Rename the component to `HumanTaskManagerComponent`.

Next, we need to implement the HumanTaskManagerComponent:

1. Right-click the **HumanTaskManagerComponent** component and select **Generate Implementation** → **Java** as shown in Figure 8-46.
2. When prompted for a package name, click **New Package,** type `com.ibm.websphere.task.sca`, and select that package from the list.

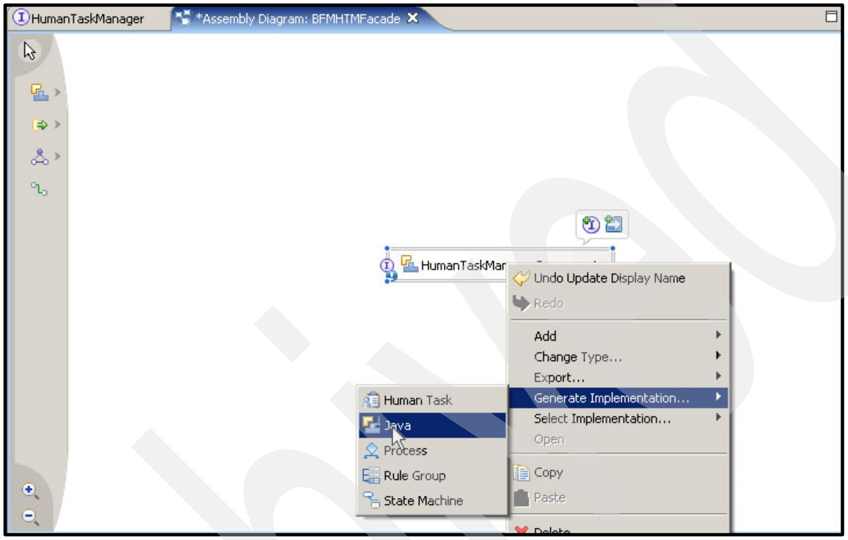


Figure 8-46 HumanTaskManagerComponent implementation

WebSphere Integration Developer will programmatically create empty methods for all methods of the HumanTaskManager interface. Now we have to implement each method, create the initHTM method, and add some private variables as shown in Example A-1 on page 368.

After saving the class, the component is complete. The Web service binding can then be generated for it. To generate the Web service binding:

1. In the Assembly editor, right-click the component, and select **Export** → **Web Service Binding**.
2. To respond to the message about programmatically generating a WSDL file, click **Yes.**
3. For the transport, select **soap/http** and click **OK.**

The service is nearly complete. The remaining item to add is the EJB reference for the Java component to reach the Human Task Manager EJB. Currently, WebSphere Integration Developer does not have a direct way of adding this reference for components, and you must use the J2EE tools instead.

These are the steps to add the EJB reference for the Java component to reach the Human Task Manager EJB:

1. Select **Window** → **Show View** → **Other.**
2. In the list of views, expand **Basic** and select **Navigator**. Click **OK**.

> **Note:** We recommend that before editing the EJB Deployment Descriptor, uncheck **Project** → **Build Programmatically** and execute a clean build through **Project** → **Clean** → **Clean All Projects**.

Find the generated EJB project, whose name is based on the name of the module holding the component. Expand **BFMHTMFacadeEJB** → **ejbModule** → **META-INF** and open `ejb-jar.xml`. (If the file is opened in a text editor, you might need to enable the Advanced J2EE capabilities in the Preferences under **Workbench** → **Preferences,** expand **Workbench**, select **Capabilities,** and check **Advanced J2EE**).

To edit the EJB Deployment Descriptor:

1. In the EJB Deployment Descriptor editor, select the **References** tab.
2. Select **Module** and click **Add**.
3. Select **EJB reference** and click **Next**.
4. In the Add EJB Reference window, select **Enterprise Beans not in the workspace**.
5. For the Name, type `ejb/LocalHumanTaskManagerHome`
6. For the Ref Type, select **Local.**
7. For the Local home interface, browse or type: com.ibm.task.api.LocalHumanTaskManagerHome
8. For the Local interface, browse or type: com.ibm.task.api.LocalHumanTaskManager
9. Click **Next** and click **Finish**.
10. For the WebSphere Binding information for the JNDI name, type: com/ibm/task/api/HumanTaskManagerHome
11. Save and close the file. (Note: If you make any changes in the Assembly editor, you might need to recreate this value because the builders in WebSphere Integration Developer will recreate the EJB project in preparation of deployment.)

The service for the Human Task Manager is complete and ready to receive requests from Web service clients to create and start tasks when we deploy the application and your application, which contains a Human Task template.

Test HumanTaskManager

Next, we test the BFMHTMFacade module that we just created. First, we test the module itself by right-clicking **BFMHTMFacade** module and select **Test** → **Test Module**.

If you already have a human task deployed and have created some tasks, then you can use the getTasks method to test the component. If this returns a result set successfully, then you are ready to test the Web service:

1. Go to the Navigator and navigate to:
`BFMHTMFacadeEJB\ejbModule\META-INF\wsdl\HumanTaskManagerComponentExport_HumanTaskManagerHttp_Service.wsdl`
2. Right-click the file name and select **Web Services** → **Test with Web Services Explorer**.
3. Select the **getTasks** method and enter the parameters. After a successful test, we can now create a Web services client.

Creating a Web service client portlet

With Rational Application Developer, you can create the portlet that will display a list of human tasks from the process server using the Web service and classes that were created by the Web service client.

First create a new portlet project. In this case, we used the IBM portlet but you can also use a JSR168 portlet but some of the API calls are slightly different.

To create a new portlet project:

1. In the Web perspective of Rational Application Developer Tools, select **File** → **New** → **portlet Project**.
2. Enter a name for the portlet, for example, `HumanTaskWeb`, and make sure you select **WebSphere Portal V5.1**. Click Next.
3. Select the **Basic Portlet**, click Next three times until you get to the EventHandling options.
4. Under **Portlet message event**, check both **Add message listener** and **Add message sender portlet sample**. This will allow the portlet to listen to messages which you will need later when a message gets sent from the Root Cause portlet to this portlet to take an action. We could also implement this using Click-to-Action, but in this case it is easier to illustrate the concept with just the message listener. Last, click **Finish.**

Next we need to enable the portlet for Web services. It is easiest to copy the wsdl files from the HumanTaskManager project into the portlet project. Copy BFMHTMFacadeEJB\ejbModule\META-INF\wsdl folder to HumanTaskWeb\WebContent\WEB-INF.

If we want to point the Web service to a different server, then we should modify the wsdl file. We can do that by opening the wsdl file in Rational Application Developer and expand the service tree until we see the SOAP address. In the property sheet, we can change the location as shown in Figure 8-47.

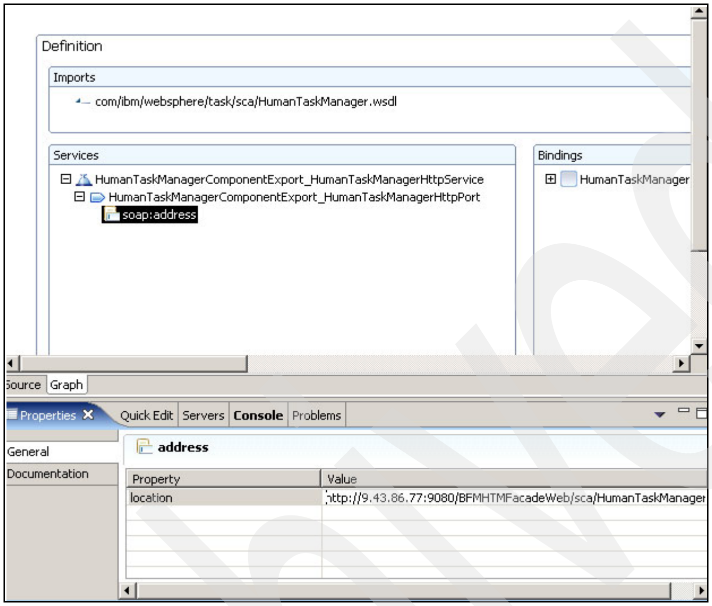


Figure 8-47 WSDL file editor

Rational Application Developer allows you to create a Web service client for your portlet project. The wizard will create all necessary classes and some test jsp pages. To create a new Web service client, follow these steps:

1. Select: **File** → **New** → **Other** → **Web Services** → **Web Service Client**.
2. As shown in Figure 8-48 on page 337, use the default settings and click **Next.**
3. Browse to the wsdl file in the directory shown below and click **Next**: WebServiceProject\WebContent\WEB-INF\wsdl\HumanTaskManagerComponentExport_HumanTaskManagerHttp_Service.wsdl
4. Last, select Client Type: **Web**, then type the current portlet project name that we want to enable for the Web service. For example, Project: `WebServiceProject` and EAR Project: `WebServiceProjectEAR`. Then, click **Finish**.

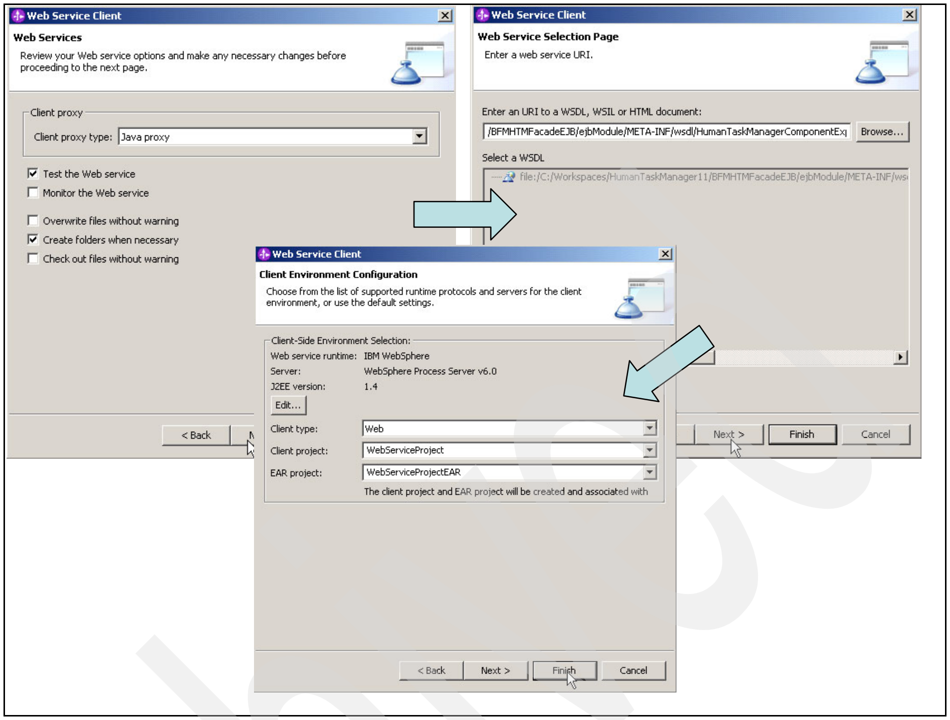


Figure 8-48 Create a Web Service Client

Now we are ready to create the user interface for the portlet.

1. Open the view jsp, for example:

   HumanTaskWeb\WebContent\humantaskweb\jsp\html\HumanTaskWebPortletView.jsp

2. First, we need to initialize the proxy by adding these two lines of code, as shown in Example 8-9, right beneath the initialization of the portlet bean, for example, HumanTaskWebPortletSessionBean.

Example 8-9 Proxy

```
BFMHTMLibrary.HumanTaskManagerProxy sampleHumanTaskManagerProxyid = new
BFMHTMLibrary.HumanTaskManagerProxy();

request.getSession().setAttribute("sampleHumanTaskManagerProxyid",sampleHumanTa
skManagerProxyid);
```

Next we need to build the table of tasks. For this, we can use the utility classes that the Web service client created. In this case, we use the getTasks method as shown in Example 8-10 on page 338. Once we have a task, we can get the *tkiid,* which is the unique identifier of that task, and that allows us to get the inputMessage using the getInputMessage method. We specifically query the `ProductCategoryAPIn` property that we specified when we created the human task on the process server, because this will give us the product group id that we will need to submit to the business intelligence portlets.

Example 8-10 Human Task list for the portlet

```
<table cellpadding="5" cellspacing="0" style="border:1px solid black">
   <tr>
      <td style="border:1px solid black;background-color:gray;color:white;">
         <STRONG>DATE</STRONG>
      </td>
      <td style="border:1px solid black;background-color:gray;color:white;">
         <STRONG>ALERT MESSAGE</STRONG>
      </td>
      <td style="border:1px solid black;background-color:gray;color:white;">
         <STRONG>STATE</STRONG>
      </td>
   </tr>
<%
try {
   String whereClause = null;
   String orderBy = null;
   Integer skipTuplesInteger  = new Integer(0);
   Integer thresholdInteger  = new Integer(0);
   BFMHTMLibrary.QueryResultTask getTasks=
      sampleHumanTaskManagerProxyid.getTasks(whereClause,orderBy,
         skipTuplesInteger,thresholdInteger);
   if(getTasks!= null){
      // Get tasks from result set
      BFMHTMLibrary.Task[] tasks = getTasks.getTasks();
      for(int i=0;i<tasks.length;i++) {
         BFMHTMLibrary.Task task = tasks[i];
         String inputMessage = null;
         // if the task is a participating task get the inputMessage
         if(task.getKind().intValue()==105) {
            inputMessage = sampleHumanTaskManagerProxyid.getInputMessage(
               task.getTkiid(),
               "ProductCategoryAPIn");
%>
<tr>
   <td style="border:1px solid black"><%=task.getActivationTime().getTime()
%></td>
   <td style="border:1px solid black"><a href="
      <portletAPI:createURI>
         <portletAPI:URIAction name='<%=HumanTaskWebPortlet.FORM_ACTION%>'/>
         <portletAPI:URIParameter name="inputMessage" value="<%=inputMessage
%>"/>
```

```
         <portletAPI:URIParameter name="tkiid" value="<%=task.getTkiid()%>"/>
      </portletAPI:createURI>
      ">High returns for <%=inputMessage %></a>
   </td>
   <td style="border:1px solid black"><%=getStateName(task.getState()) %></td>
</tr>
<%
         }
      }
    }

} catch (Exception e) {
%>
exception: <%= e %>
<%
return;
}
%>
</table>
<%!
   // Utility method to show state of task
   public String getStateName(Integer state) {
      if(2==state.intValue())
         return "Ready";
      if(8==state.intValue())
         return "Claimed";
      return state.toString();
   }
%>
```

In order to store generic variables in the portlet session bean, we add a hashMap and a method to store name value pairs. Open: HumanTaskWeb\JavaSource\humantaskweb\HumanTaskWebPortletSessionBean.java

Add the lines of code as shown in Example 8-11.

Example 8-11 Portlet Session Bean generic HashMap

```
private HashMap hashMap = new HashMap();

public void setValue(String param,String value) {
   this.hashMap.put(param,value);
}

public String getValue(String param) {
   return (String)this.hashMap.get(param);
}
```

Now open the portlet controller:
HumanTaskWeb\JavaSource\humantaskweb\HumanTaskWebPortlet.java

Add the static variables, as shown in Example 8-12, to the top of the class.

Example 8-12 Static variables for portlet controller

```
public static final String CURRENT_TKIID= "humantask.CurrentTkiid";
public static final String TKIID = "tkiid";

// Action related variables
public static final String MESSAGE= "humantask.message";
public static final String TASK_COMPLETE_ACTION =
      "humantask.TaskCompleteAction";
public static final String CHANGE_SHIPPER_PREFIX = "ChangeShipper:";
public static final String INPUT_MESSAGE= "inputMessage";
```

Now we have to handle the event when the user clicks a task. In the actionPerformed() method, get the *tkiid* of the task and the inputMessage which is the product category id. Store the *tkiid* in the session bean for later reference when we want to complete the task. You also claim the task with the *tkiid* and send a message with the inputMessage to the other portlets on the page, as shown in Example 8-13. This will allow you later to pick up the product category with the Alphablox portlets and slice the data dynamically.

Example 8-13 Task Claim Action

```
if( FORM_ACTION.equals(actionString) ) {
   // Set form text in the session bean
   String inputMessage = request.getParameter("inputMessage");
   String tkiid = request.getParameter(TKIID);
   BFMHTMLibrary.HumanTaskManagerProxy sampleHumanTaskManagerProxyid =
      (BFMHTMLibrary.HumanTaskManagerProxy)request.getSession().
         getAttribute("sampleHumanTaskManagerProxyid");
   try {
      System.out.println("Claim:"+tkiid);
      sessionBean.setValue(CURRENT_TKIID,tkiid);
      sampleHumanTaskManagerProxyid.claimTask(tkiid);
   } catch (RemoteException e) {
      e.printStackTrace();
   } catch (ClaimTask_faultMessageMsg e) {
      e.printStackTrace();
   }
   // Send message to other portlets
   PortletMessage message = new DefaultPortletMessage(inputMessage);
   getPortletConfig().getContext().send(null,message);
}
```

You can also add a second handler for the action part of the portlet that will allow you to complete the task. For this, add a second handler to the actionPerformed() method.

Example 8-14 Task Complete Action

```
if( TASK_COMPLETE_ACTION.equals(actionString) ) {
   String tkiid = request.getParameter(TKIID);
   BFMHTMLibrary.HumanTaskManagerProxy sampleHumanTaskManagerProxyid =
      (BFMHTMLibrary.HumanTaskManagerProxy)request.getSession().
         getAttribute("sampleHumanTaskManagerProxyid");
   try {
      // Set session variables to null
      sessionBean.setValue(MESSAGE,null);
      sessionBean.setValue(CURRENT_TKIID,null);
      sampleHumanTaskManagerProxyid.completeTaskWithMessage(tkiid,null);
   } catch (RemoteException e) {
      e.printStackTrace();
   } catch (CompleteTaskWithMessage_faultMessageMsg e) {
      e.printStackTrace();
   }
}
```

In order to receive messages from other portlets, a method, messageReceived(), is already implemented in the portlet controller based in the preferences when you created the portlets. In this case, you want to receive a message about the name of the shipper that should be changed and the information will be passed by an Alphablox portlet. To implement this, store the message in the session bean as shown in Example 8-15 and then process the value on the jsp page.

Example 8-15 messageReceived

```
HumanTaskWebPortletSessionBean sessionBean =
                                   getSessionBean(event.getRequest());
sessionBean.setValue(MESSAGE,messageText);
```

Last, we need to add the user interface for the action to the portlet view jsp. Add the code to the bottom of the jsp page, as shown in Example 8-16. This will create a form that will show the shipper to be changed and allow the user to enter a new shipper.

Example 8-16 Human Task Action user interface

```
<%
String message = sessionBean.getValue(HumanTaskWebPortlet.MESSAGE);
String tkiid = sessionBean.getValue(HumanTaskWebPortlet.CURRENT_TKIID);
if(tkiid!=null && message!=null &&
         message.startsWith(HumanTaskWebPortlet.CHANGE_SHIPPER_PREFIX)) {
```

```
   String shipper = message.substring(message.indexOf(
      HumanTaskWebPortlet.CHANGE_SHIPPER_PREFIX)+
         HumanTaskWebPortlet.CHANGE_SHIPPER_PREFIX.length());
%>
<h3>Change Shipper</h3>
Please enter the new shipper.
<FORM method="POST"
   action="<portletAPI:createURI>
            <portletAPI:URIAction
               name='<%=HumanTaskWebPortlet.TASK_COMPLETE_ACTION%>'/>
         </portletAPI:createURI>">
    <INPUT class="wpsEditField" name="<%=HumanTaskWebPortlet.TKIID %>"
         value="<%=tkiid%>" type="hidden"/>
    <LABEL class="wpsLabelText" for="currentShipper">
      Current Shipper: 
   </LABEL>
   <INPUT class="wpsEditField" name="currentShipper" value="<%=shipper%>"
      type="text"/>
       
    <LABEL class="wpsLabelText"   for="newShipper">New Shipper</LABEL>
    <INPUT class="wpsEditField"  name="newShipper" value="" type="text"/><BR>
    <INPUT class="wpsButtonText" name="
      <portletAPI:encodeNamespace value='<%=HumanTaskPortlet.SUBMIT%>'/>"
         value="Complete" type="submit"/>
</FORM>
<%
}
sessionBean.setValue(HumanTaskWebPortlet.MESSAGE,null);
%>
```

Note: It is possible that you might have problems deploying the portlet, because the path to the wsdl file is too long for a Windows system to handle. If so, you need to shorten the names of the files that Rational Application Developer created.

8.6.2 Alphablox portlets: Looking for the root cause

In this section, we describe how to implement a custom Alphablox portlet and how it can cooperate with other portlets. In Figure 8-49 on page 343, we illustrate the Alphablox portlets and their interactions. In the last section, we described the My Tasks portlet and how it sends messages with the product group to the other portlets. Now we describe how the Alphablox portlets will receive the parameters, how the Alphablox portlets are built, and how they cooperate with each other.

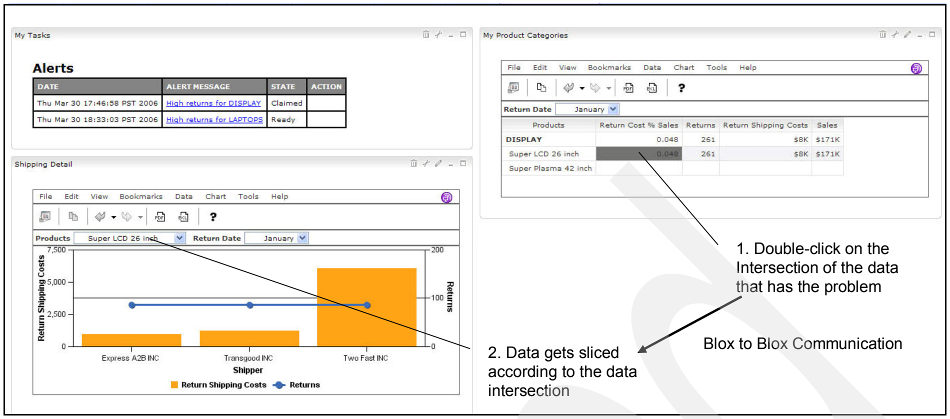


Figure 8-49 Alphablox portlets

Basic Alphablox portlet setup

We built the custom Alphablox portlet using Rational Application Developer V6.0.1, before we started building the portlet to make sure the Alphablox Rational Application Developer plug-in is installed and Alphablox is installed into the Rational Application Developer test environment. For more information, refer to the Information Center:

`http://publib.boulder.ibm.com/infocenter/ablxhelp/v8r4m0/index.jsp?topic=/com.ibm.db2.abx.gst.doc/abx-t-start-22.html`

Next, create a new portlet project. In this case, we use the IBM portlet. You can also use a JSR168 portlet, but some of the API calls are slightly different. In the Web perspective of Rational Application Developer, select **File → New → Portlet Project.** The steps to create a new portlet project are:

1. Enter a name for the portlet, for example, `Alphablox`, and make sure you select **WebSphere Portal V5.1.**
2. Click **Next** and select the **Basic Portlet,** then click **Next** and check **DB2 Alphablox Content** under Features. This will include the Alphablox libraries into the project for java completion and tag completion.
3. Then click Next two times and you get to the Event Handling options. Under **Portlet message event,** check both **Add message listener** and **Add message sender portlet sample.** This will allow the portlet to listen to messages, which you will need when a message gets sent from the My Tasks portlet to this portlet, to slice the data according to the product group.
4. Click **Next** two times and you get to the Miscellaneous window, check **Add edit mode**. Use the edit mode to configure the Alphablox portlet. Last, click **Finish**.

If you forget to check the **DB2 Alphablox Content** option, then you can right-click the project and select **Properties**. In the Properties dialog, select **Web Project Features** and check **DB2 Alphablox Content**.

To create a basic Alphablox portlet, you need to add three parts to the portlet view jsp (AlphabloxportletView.jsp):

1. Alphablox tag libraries to the top of the jsp page. This will allow the servlet engine to resolve the Alphablox tags.

```
<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="bloxuitld" prefix="bloxui"%>
<%@ taglib uri="bloxportlettld" prefix="bloxportlet"%>
```

2. The header tag, which is required, because it will include the JavaScript client API and reference to the style sheets.

```
<blox:header/>
```

3. A visible blox, for example, `PresentBlox`. We will separate the DataBlox from the PresentBlox and have a separate Display blox. This will make the portlet slightly more flexible for extensions of the portlet.

```
<blox:data
    id="dataBlox"
    dataSourceName="CubeViews"
    query="select from [QCC]"/>

<blox:present
   id="presentBlox"
   visible="false">
   <blox:data bloxRef="dataBlox"/>
</blox:present>

<DIV style="margin: 6px">
<table width="100%" cellpadding="0" cellspacing="0">
   <tr>
      <td>
         <blox:display
            bloxRef="presentBlox"
            height="500"
            width="100%" />
      </td>
   </tr>
</table>
</DIV>
```

4. At this point, you can test the portlet. And, assuming you have the QCC sample database setup and created, the data sources in the Alphablox admin pages. You will see a basic blox in the portlet with a default query.

Next we add some bookmark capabilities. Eventually, this will allow us to create a report in the portlet edit mode, save it as a bookmark, and then retrieve it in the view mode persistently.

Alphablox uses *bookmarks* to save and load report definitions that contain all the properties of a blox, for example, query, data source, chart type, and page filters. The bookmark is an exact replica of all the property values of the blox and its nested blox. It is important to know that when a bookmark gets saved, it will only save the properties that have changed due to interactions with the blox and not properties that were already set by the lag libraries or by the jsp through the Java API. The bookmarks are saved in the Alphablox repository. They can be loaded and saved directly from one of the blox or through the BookmarkBlox.

To add basic bookmark capabilities, we add the following four parts to the portlet:

1. Add a bloxName/bookmark name variable to the top of the JSP. This variable will contain the bookmark name that we want to retrieve:

   ```
   String bloxName = "myBlox";
   ```

2. Add a BookmarkBlox right before the DataBlox. The BookmarkBlox gives us access to the bookmark API, so that we can retrieve and save bookmark properties without loading the bookmark.

   ```
   <blox:bookmarks id="bookmarksBlox"/>
   ```

3. Add a bookmarkFilter attribute to the PresentBlox. This will make sure that all bookmarks get saved and stored under the blox name presentBlox, instead of the actual PresentBlox name. That makes it easier to save and restore bookmarks.

   ```
   bookmarkFilter=",name=presentBlox"
   ```

4. Add a JSP scriplet that will load the bookmark:

   ```
   <%
      // Load Bookmark code
       if(bloxName!=null) {if(bookmarksBlox.bookmarkExists(
         bloxName,presentBlox.getApplicationName(),
         "",
         presentBlox.getBloxName(),
         Bookmark.PUBLIC_VISIBILITY,
         presentBlox.getBookmarkFilter())) {
            Bookmark bookmark =
               bookmarksBlox.getBookmark(bloxName,
               presentBlox.getApplicationName(),
               "",
               presentBlox.getBloxName(),
               Bookmark.PUBLIC_VISIBILITY,
               presentBlox.getBookmarkFilter());
            // Bookmark Properties can be overwritten on load
   ```

```
                BookmarkProperties bookmarkProperties =
                bookmark.getBookmarkPropertiesByType(Bookmark.PRESENT_BLOX_TYPE);
                bookmarkProperties.setProperty("dataLayoutAvailable","false");
                presentBlox.loadBookmark(bookmark);
            }
        }
%>
```

Now that we have explained how to enable a blox to use bookmarks, we also need to do the same thing for the edit view of the portlet, because the edit view will serve us as the report builder user interface and save the bookmark. We also need to add some parameters to the edit mode that allow us to have a more generic, configurable portlet.

We will add the following parameters:

- **bloxName**

  Based on the bloxName, which has to be unique for each portlet, the DataBlox and PresentBlox name will be created, as PresentBlox and DataBlox names need to be unique on a single portal page in order to coexist. The bloxName will also be equal to the bookmark name which stores all the configuration parameters of the DataBlox, PresentBlox, and the nested blox.

- **portletTitle**

  The portletTitle property will change the title of the portlet. The property will be used in the portlet controller where we implement a PortletTitleListener, which allows portlet titles to be changed.

- **height**

  The height attribute will set the height attribute of the PresentBlox or the DisplayBlox, which renders the PresentBlox. The height will ultimately also determine the height of the portlet. This is necessary because the height cannot be specified for a portlet; otherwise, the PresentBlox would not show up correctly assuming 100% sizing. The results might differ slightly by browser type.

- **targetPresentBlox**

  The targetPresentBlox is a property that is used for the Blox-to-Blox communication. The administrator specifies the bloxName of the portlet with which this portlet should interact when a data cell gets double-clicked.

- **pageTarget**

  The pageTarget property has the values of `true` or `false` and tells the portlet if it should receive Blox-to-Blox communication from other portlets that change the page filter.

- **rowTarget**

The rowTarget property has the values of `true` or `false` and tells the portlet if it should receive Blox-to-Blox communication from other portlets that change the row set.

- **portletLink**

  The portletLink property has the value of `true` or `false` and tells the portlet if a portlet message using the WebSphere Portal API should be sent to other portlets on the page by double-clicking a cell or chart component.

Now that the edit view has been defined and view mode is enabled for bookmarks, we need to make the view mode dynamic to pick up the parameters from the edit mode, especially the bloxName, so that more than one of these portlets can be on a single portlet page. To do that, refer to Example 8-17.

Example 8-17 Making the view mode dynamic

```
<%
   // Parameters for edit mode
   PortletData portletData = portletRequest.getData();

   String bloxName = (String)portletData.getAttribute("bookmarkName");
   boolean pageTarget =
      "true".equals(getParam(portletData.getAttribute("pageTarget")));
   boolean rowColTarget =
      "true".equals(getParam(portletData.getAttribute("rowColTarget")));
   boolean portletLink =
      "true".equals(getParam(portletData.getAttribute("portletLink")));
   String height = (String)portletData.getAttribute("height");
   // Set dynamic height parameters
   if(height==null || height.equals(""))
      height = "300";
   // Message if there is no bookmark available
   if(bloxName==null) {
      out.write("<b>Please go to the Edit Mode to set up the report.</b>");
      return;
   }
   // Dynamic blox names
   String dataBloxName = bloxName+"_data";
   String presentBloxName = bloxName+"_present";
%>
<%!
// Utility method to get parameters from edit mode
public Object getParam(Object value) {
   if(value==null)
      return "";
   else if(value.equals("null"))
      return "";
   return value;
}
```

```
%>
```

Currently, the DataBlox and PresentBlox have a static id attribute which is equal to the variable on the jsp page as well as the session object. Because we want to deploy more than one blox on a portal page, we need to make the session name dynamic. We can do that by using the bloxName attribute for the DataBlox and PresentBlox because it accepts dynamic variables, and the id attribute does not. We also need to change the BloxRef attribute for the nested DataBlox of the PresentBlox and the DisplayBlox as shown in Example 8-18.

Example 8-18 Dynamic PresentBlox and DataBlox names

```
<blox:data
   id=”dataBlox”
   bloxName=”<%=dataBloxName%>”
   .../>

<blox:present
   id=”presentBlox”
   bloxName=”<%=presentBloxName%>”
   ...>
   <blox:data bloxRef=”<%=dataBloxName%>”.../>
   ...
</blox:present>
....
<blox:display bloxRef=”<%=presentBloxName%>”.../>
```

Now, we have created a portlet where the PresentBlox can be configured in the edit view and persistently displayed in the view mode. We can now create the data view for both Alphablox portlets.

Process product group

In this section, we describe how to receive the message from the My Tasks portlet and process the product group to change the data dynamically.

First, we need to receive the message from the My Tasks portlet containing the product group name, we use the same method that we described in “Creating a Web service client portlet” on page 335.

Add the HashMap and the two methods getValue and setValue to the portlet session bean (for example, the AlphabloxPortletSessionBean.java). See Example A-9 on page 407. Then, add the code to the messageReceived method in the portlet controller (for example, AlphabloxPortlet.java), which will put the product group name into the portlet session bean as shown in Example 8-19 on page 349.

Example 8-19 AlphabloxPortlet messageReceived()

```
AlphabloxPortletSessionBean sessionBean = getSessionBean(event.getRequest());
sessionBean.setValue("message",messageText);
```

Next, we need to retrieve the message from the session bean and we will use the setSelectedMembers method, which is a method on the DataBlox object, to set the members on the page, row, or column. This method only works with multidimensional data sources.

The parameter for the method is an array of Member which is part of the MDBMetaData object. The code in Example 8-20 shows how to resolve the product code with the MDBMetaData, then we need to check if the dimension that relates to these members is on the page, row, or column.

Example 8-20 Change Data dynamically based on Portlet Message

```
<%
String messageText = sessionBean.getValue("message");
if(dataBlox.getMetaData() instanceof MDBMetaData) {
   // Get metadata object
   MDBMetaData meta = (MDBMetaData)dataBlox.getMetaData();
   // Get result set object
   MDBResultSet rs = (MDBResultSet)dataBlox.getResultSet();
   if(messageText!=null) {
      // check if the member name is in MDX format
      if(!messageText.startsWith("["))
         messageText = "["+messageText+"]";
      // resolve member in the metadata object
      Member member = meta.resolveMember(messageText);
      if(member!=null) {
         AxisDimension axisDimension =
rs.resolveAxisDimension(member.getDimension().getDisplayName());
         // Check on what axis the dimension of the member
         // currently is
         if(pageTarget && axisDimension!=null &&
axisDimension.getAxis().getIndex()==Axis.PAGE_AXIS_ID) {
            // For page only that member should be selected
            dataBlox.setSelectedMembers(new Member[]{member});
         }
         if(rowColTarget && axisDimension!=null &&
axisDimension.getAxis().getIndex()==Axis.ROW_AXIS_ID) {
            // For column or row select the member and its childrem
            Member[] members = member.getChildren();
            Member[] hierarchy = new Member[members.length+1];
            hierarchy[0] = member;
            for(int i=1;i<hierarchy.length;i++)
               hierarchy[i] = members[i-1];
            dataBlox.setSelectedMembers(hierarchy);
```

```
            }
        }
    }
}
%>
```

To test this functionality, you first have to enable the *Allow change Page* and *Allow change Row or Column,* then you can use the Message Sender Portlet to send the name of a member on the page, row, or column.

Blox-to-Blox communication

In this section, we show how to communicate from one blox in a portlet to a blox in another portlet. The advantage of the approach described here over the portlet message sender or the click-to-action is that it does not require a page refresh, but this only works with Alphablox portlets.

To enable Blox-to-Blox communication, we need to take the following steps:

1. Create an EventHandler.

   ```
   public class BloxToBloxEventHandler implements IEventHandler {
      PresentBlox presentBlox;
      // name of the target presentBlox(s)
      String targetPresentBlox;
      // Put all the coordinates into a Hashmap
      Hashtable memberHash = new Hashtable();


      public BloxToBloxEventHandler(PresentBlox presentBlox,
         String targetPresentBlox) throws Exception {
         this.presentBlox = presentBlox;
         this.targetPresentBlox = targetPresentBlox;
      }
   }
   ```

2. Create the type of handler that will trigger the Blox-to-Blox action, in this case, a doubleClickEventHandler inside the BloxToBloxEventHandler.

   ```
      public boolean handleDoubleClickEvent(DoubleClickEvent event)
         throws Exception {
         Component component = event.getComponent();
         // Check if double click happened on a GridCell
         if(component instanceof GridBrixCellModel) {
          GridCell gridCell = (GridCell) component;
            // Check if the cell is a data cell
          if(!gridCell.isRowHeader()&& !gridCell.isColumnHeader()) {
             getMembers();

            return true;
   ```

```
            }
        }
        return false;
       }
```

3. Get the current selected GridCell (see complete code in step 5).

```
GridCell[] gridCells = grid.getSelectedCells();
```

4. Find the Cell object in the ResultSet using the findGridBrixCell() method, which links a GridCell to a Cell object in the result set so that we can find the dimension members for that cell (see complete code in step 5).

```
Object object = grid.findGridBrixCell(rs,gridCell);

// If the Object is a data cell then cast to Cell
if(object instanceof Cell) {
   Cell cell = (Cell) object;
   ....

}
// if the object is a Tuple Member
else if(object instanceof TupleMember) {
   TupleMember tupleMember = (TupleMember)object;
   ...
}
```

5. Get the members for each dimension that relate to the selected cell and store in a HashMap.

```
public Hashtable getMembers() throws ServerBloxException {
   // Get GridBrixModel
   GridBrixModel grid = presentBlox.getPresentBloxModel().getGrid();
   // get MDB Result Set
   MDBResultSet rs = (MDBResultSet)
      presentBlox.getDataBlox().getResultSet();
   // Get all selected cells
   GridCell[] gridCells = grid.getSelectedCells();

   // loop through all selected cells
   for (int i = 0; i < gridCells.length; i++) {
      GridCell gridCell = gridCells[i];
      // get the corresponding object to the GridCell in the MDBResltSet
      Object object = grid.findGridBrixCell(rs,gridCell);

      // If the Object is a data cell then cast to Cell
      if(object instanceof Cell) {
         Cell cell = (Cell) object;
         // Get the row and column tuples for the cell
         Tuple[] tuples = cell.getTuples();
         for (int j = 0; j < tuples.length; j++) {
```

```
        Tuple tuple = tuples[j];
        TupleMember[] members = tuple.getMembers();
        for (int k = 0; k < members.length; k++) {
           TupleMember member = members[k];
           // exclude calculated members
           if(!member.isCalculatedMember()) {
              String uniqueMember = member.getUniqueName();
              // in case the member is a shared Essbase member
              // use the display name
              if(uniqueMember.indexOf("\u0001")>-1)
                 uniqueMember = member.getDisplayName();
              // Add member to hash map
              memberHash.put(member.getDimension().getDisplayName(),
                       uniqueMember);
           }
        }
     }
     // Also add all page members
     Axis pageAxis = rs.getAxis(Axis.PAGE_AXIS_ID);
     for(int j=0;j<pageAxis.getTupleCount();j++) {
        Tuple tuple = pageAxis.getTuple(j);
        for(int k=0;k<tuple.getMemberCount();k++) {
           TupleMember member = tuple.getMember(k);
           memberHash.put(
              member.getDimension().getDisplayName(),
              member.getUniqueName().substring(
                 member.getUniqueName().indexOf(".")+1));
        }
     }
     // Last get the cube name and add it to the hash map
     Cube cube = rs.getCubes()[0];
     memberHash.put("cube",cube.getName());
  }
  // if the object is a Tuple Member
  else if(object instanceof TupleMember) {
     TupleMember tupleMember = (TupleMember)object;
     if(gridCell.isRowHeader()) {
        memberHash.put(
           tupleMember.getDimension().getDisplayName(),
           tupleMember.getUniqueName());
        Axis pageAxis = tupleMember.getTuple().getAxis().
           getResultSet().getAxis(Axis.PAGE_AXIS_ID);
        for(int j=0;j<pageAxis.getTupleCount();j++) {
           Tuple tuple = pageAxis.getTuple(j);
           for(int k=0;k<tuple.getMemberCount();k++) {
              TupleMember member = tuple.getMember(k);
              memberHash.put(
                 member.getDimension().getDisplayName(),
                 member.getUniqueName().substring(
```

```
                        member.getUniqueName().indexOf(".")+1));
                }
            }
        }
    }
    }
    return memberHash;
}
```

6. Find the target blox in the session object (see complete code in step 7).

```
BloxContext bloxContext = presentBlox.getBloxContext();
PresentBlox blox = (PresentBlox)bloxContext.getBlox(bloxName);
```

7. Apply selected members from the source blox using setSelectedMembers().

```
try {
   StringTokenizer stringTokenizer = new
   StringTokenizer(this.targetPresentBlox,",");
   // Get the BloxContext to find the other blox
   BloxContext bloxContext = presentBlox.getBloxContext();
   while(stringTokenizer.hasMoreTokens()) {
      String token = stringTokenizer.nextToken();
      String bloxName = token+"_present";
      // Check for the targetPresentBlox if member exists
      // and on what axis
      PresentBlox blox = (PresentBlox)bloxContext.getBlox(bloxName);
      MDBResultSet rs = (MDBResultSet) blox.getDataBlox().getResultSet();
      MDBMetaData meta = (MDBMetaData) blox.getDataBlox().getMetaData();
      if(blox!=null) {
         Enumeration dimensionNames = memberHash.keys();
         while(dimensionNames.hasMoreElements()) {
            String dimName = (String)dimensionNames.nextElement();
            String memberName = (String) memberHash.get(dimensionName);
            Member member = meta.resolveMember(memberName);
            AxisDimension axisDimension = rs.resolveAxisDimension(dimName);
            // Consider only members that are on page
            if(axisDimension!=null &&
               axisDimension.getAxis().getIndex()==Axis.PAGE_AXIS_ID) {
               if(member!=null)
                  blox.getDataBlox().setSelectedMembers(
                     new Member[]{member});
            }
         }
      }
   }
} catch(Exception e) {
   MessageBox.message( component, "Error", "Error:"+e.getMessage());
}
```

8. Add EventHandler to the PresentBloxModel inside the PresentBlox tag, so that the EventHandler only gets added on initialization.

```
// Blox to Blox EventHandler
String targetPresentBlox =
   (String)portletData.getAttribute("targetPresentBlox");
if(targetPresentBlox!=null)
   presentBlox.getPresentBloxModel().addEventHandler(
      newBloxToBloxEventHandler(presentBlox,targetPresentBlox));
```

Portlet link to close the loop

We have been alerted about the returns problem, analyzed the problem by product, and identified the shipper we want to change in the root cause analysis. Now we need to take the action to change the shipper. For that, we need to collect information from the Alphablox portlet and send a message, using cooperative portlets, back to the My Tasks portlet which will be able to complete the task and start a new process to actually change the shipper (see Figure 8-50).

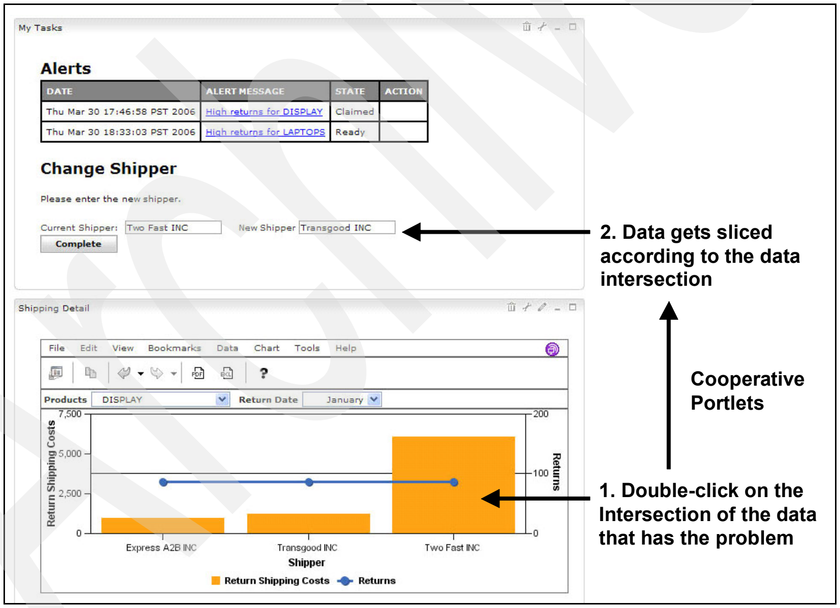


Figure 8-50 Close the loop and complete the task

To connect the chart of the Alphablox portlet to the My Tasks portlet, we use a PortletLinkEventHandler to get the information from the chart and then invoke a portlet message to the My Tasks portlet. Again, we could also use a different method for cooperative portlets, but this is the easiest method to illustrate. We could also invoke the portlet link from a grid similar to the BloxToBloxEventHandler.

To create the portlet link, perform the following steps:

1. Add the PortletLink tag nested to the PresentBlox tag. This creates a PortletLink object that allows us to call a portlet action URL from the Alphablox EventHandler with some parameter value pairs.

```
<bloxportlet:actionLinkDefinition action="<%=actionName%>">
    <bloxportlet:parameter name="<%=paramName%>" />
</bloxportlet:actionLinkDefinition>
```

2. Create an Event Handler, for example, a `PortletLinkEventHandle`. We will only pass in the PresentBlox because the PortletLink object can be retrieved from the PresentBlox.

```
public class PortletLinkEventHandler implements IEventHandler {
   // Source PresentBlox
   PresentBlox presentBlox;
   // Portlet Link object specified as nested blox of the PresentBlox
   PortletLink portletLink;

   public PortletLinkEventHandler(PresentBlox presentBlox) {
      this.presentBlox = presentBlox;
      this.portletLink =
         presentBlox.getPortletLink(AlphabloxPortlet.PORTLET_LINK);
   }
}
```

3. Create the type of handler that will trigger the portlet link action, in this case, a doubleClickEventHandler inside the PortletLinkEventHandler. Note that we return false as the default, that means that other EventHandler will still be executed.

```
public boolean handleDoubleClickEvent(DoubleClickEvent event)
      throws Exception {
      Component component = event.getComponent();

      return false;
}
```

4. Get the selected chart component. We also return `true` so that no other EventHandlers will be executed, for example, which is the default behavior of the double-click.

```
// Check if clicked component is a Chart
if(component instanceof Chart) {
   Chart theChart = (Chart) event.getComponent();
   // get the selected component in the chart
   ChartComponent chartComponent = theChart.getSelectedChartComponent();
   // Check if the selected component is a single data
```

```
    // series, for example, a bar
    if (chartComponent instanceof SingleValueDataSeries) {
       ChartBrixModel cbModel=presentBlox.getPresentBloxModel().getChart();
       String memberName = null;
       SingleValueDataSeries series=(SingleValueDataSeries) chartComponent;
       // return true so that no other action will be performed
       return true;
    }
}
```

5. Get selected x-Axis member (for example, `shipper`). In Alphablox 8.4 and later, a native ChartDataPoint can be retrieved from the data series which will allow us to get unique member name for the series (`legend`), group (`x-axis`) and filters. For versions prior to Alphablox 8.4, we need to get the axis label, but that might not give us the correct result in all cases. When we have the unique members, we will resolve them in the Metadata object to get the display name which we want to pass on.

```
int selectedIndex = series.getSelectedIndex();
// Get the Navtive Data Point this
ChartDataPoint nativeDataPoint = series.getNativeDataPoint(selectedIndex);
// Get series members, i.e. legend
String[] seriesMembers = cbModel.getUniqueSeriesMembers(nativeDataPoint);
// Get group members, i.e. x-axis
String[] groupMembers = cbModel.getUniqueGroupMembers(nativeDataPoint);
// Get filter members
String[] filterMembers = cbModel.getUniqueFilterMembers();
// Get the first member on the x-axis, for example shipper
// in this case we know there is only one dimension on the x-axis
memberName = groupMembers[0];
// resolve the member in the MetaData object to get the
// display name
MetaData metaData = presentBlox.getDataBlox().getMetaData();
if(metaData instanceof MDBMetaData) {
   MDBMetaData mdbMetaData = (MDBMetaData)metaData;
   Member member = mdbMetaData.resolveMember(memberName);
   memberName = member.getDisplayName();
}
```

6. Set the value on the portlet link and then forward the URL. We will put a prefix of “ChangeShipper:” to the message so that only the My Tasks portlet will pick up the message. This could also be done with portlet wiring, which allows a message to be sent only to particular portlets. Also, once we have set the shipper name on the portlet link, the PortletLink has a method to return a JavaScript which sends an action call to the portlet controller with the shipper name in this case. To invoke the JavaScript method, we use the method of the blox model getDispatcher().sendClientCommand(...)

```
portletLink.setParameterValue(AlphabloxPortlet.PORTLET_LINK_PARAM,
   "ChangeShipper:"+memberName);
cbModel.getDispatcher().sendClientCommand(portletLink.getLinkCall());
```

7. Add PortletLinkEventHandler to the PresentBloxModel.

```
// Add PortletLinkEventHandler that handles Click to Action
if(portletLink)
   presentBlox.getPresentBloxModel().addEventHandler(
      new alphablox.event.PortletLinkEventHandler(presentBlox));
```

Clean up the User Interface

Last, we clean up the view mode of the portlet by disabling some of the toolbar buttons that are not necessary to expose to users and disable the menu bar.

The menu can be disabled by setting the attribute *menuVisible="false"* on the PresentBlox tag. To disable one of the toolbars or individual buttons, we use the bloxui tag, as shown in Example 8-21. The bloxui tags should be placed after the load bookmark function; otherwise, the bookmark can potentially overwrite the toolbar settings.

Example 8-21 Blox UI tag to disable toolbars

```
<% // --> Set Toolbar buttons %>
<bloxui:toolbar name="<%=ModelConstants.NAVIGATION_TOOLBAR%>"
      title="" visible="false"/>
<bloxui:toolbar name="<%=ModelConstants.STANDARD_TOOLBAR%>" title="">
   <bloxui:toolbarButton name="<%=ModelConstants.BOOKMARK_LOAD%>"
         title="" visible="false"/>
</bloxui:toolbar>
<% // <-- End Set Toolbar buttons %>
```

Test the Alphablox Portlet

The portlet can either be tested through Rational Application Developer or deployed to a WebSphere Portal Server. After the portlet is rendered, it will display the following message for the first time:

```
“Please go to the Edit Mode to set up the report.”
```

This is because no bookmark has been defined for this Alphablox portlet.

We now have to put the portlet into edit mode.

First, select a bookmark name which will be the internal report name. After that, we can configure the portlet using the following parameters:

- **Bookmark Name**

  This is the name of the bookmark in the Alphablox repository. Using the Change button, the blox can be pointed to a different bookmark.

- **Portlet Title**

  The property allows you to change the portlet title.

- **Height**

  This property specifies the height of the PresentBlox.

- **Blox to Blox List**

  This property specifies a comma delimited list of blox which are the target of the Blox-to-Blox communication from this portlet.

- **Allow change Page**

  This property specifies whether this portlet will allow changes to the page filters when it receives a message from another portlet. This property is required to be enabled for the Root Cause portlet.

- **Allow change Row or Column**

  This property specifies whether this portlet will allow changes to the row or column set when it receives a message from another portlet. This property is required to be enabled for the Analyze portlet.

- **Enable Portlet Links**

  This property specifies whether a portlet link will be triggered from the chart as a portlet message to the other portlets. For the Root Cause portlet, this property needs to be enabled to communicate with the My Tasks portlet.

Last, it also shows a PresentBlox that allows us to select a data source and a cube. Then, we can create the reports by dragging the dimensions on the different axes. After we save the portlet data, we can test the portlet using the Message Sender portlet to see how it receives messages and sends messages.

8.7 The solution execution

In this section, we describe how to execute the case study solution that we have developed in this chapter.

8.7.1 Returning products

The case study scenario starts when a call is received from a customer who wants to return a purchased product. The Call Center operator receives the call and documents the required information for the product return process.

That call initiates an instance of the GoodsReturnProcess. This process, as explained in 7.3, "The case study returns process flow" on page 251, initiates analysis for the call regarding the warranty period and other pertinent process information.

Initiating the process is done using a custom client that connects to WebSphere Process Server. However, in this sample implementation, we are using *BPC Explorer*, which is the built-in client that comes with WebSphere Process Server.

To create an instance of the GoodsReturnProcess:

1. Go to the URL:

   `http://process_server_machine_ip:9080/bpc`

   Which is the window with the process templates.
2. On the left pane, click **My Process Templates**.
3. A list of all the deployed processes displays. There are two processes depicted. One, the *GoodsReturnProcess*, is the one used to create an instance. The second is the *ProbelmEvaluationProcess*, which is the corrective action process. They are depicted in Figure 8-51 on page 360.

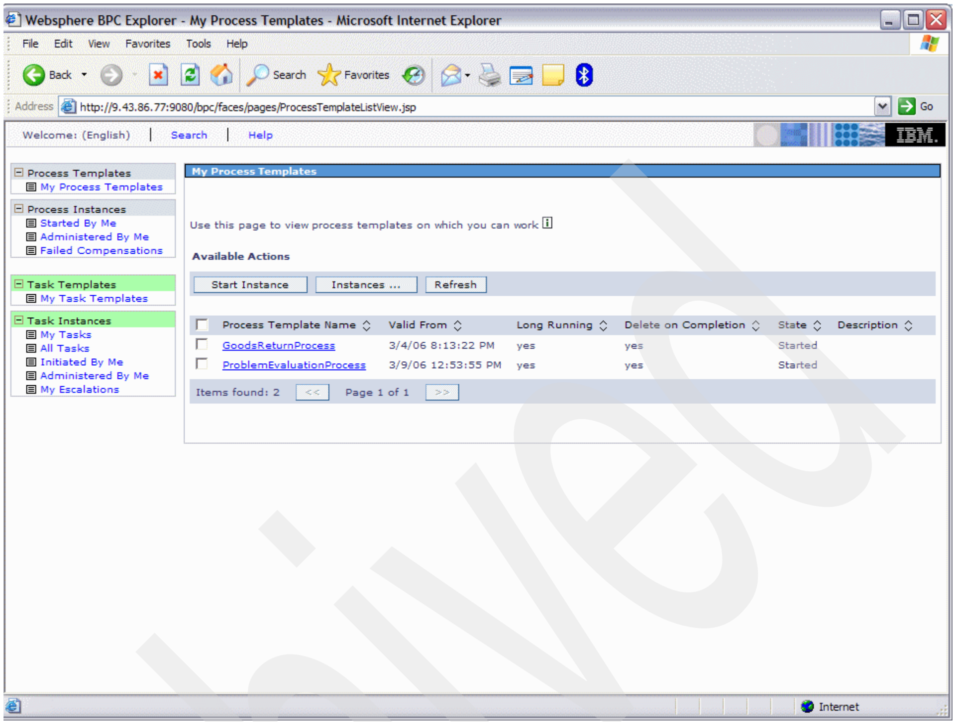


Figure 8-51 Deployed process templates

4. Check the check box to the right of *GoodsReturnProcess* and click **Start Instance** button at the top of the page.
5. A page shows that asks for the data required to start a new instance of *GoodsReturnProcess*, see Figure 8-52 on page 361.
6. Enter the relevant information. Two fields are important here, the **Product Category**, and **Shipper Name**, because we are interested in monitoring the number of returns for each product category for the current day. *Shipper Name* is important because we are analyzing the number of returns for each shipper, and want to see whether or not the problem is related to a specific shipper.
7. Click **Submit**.

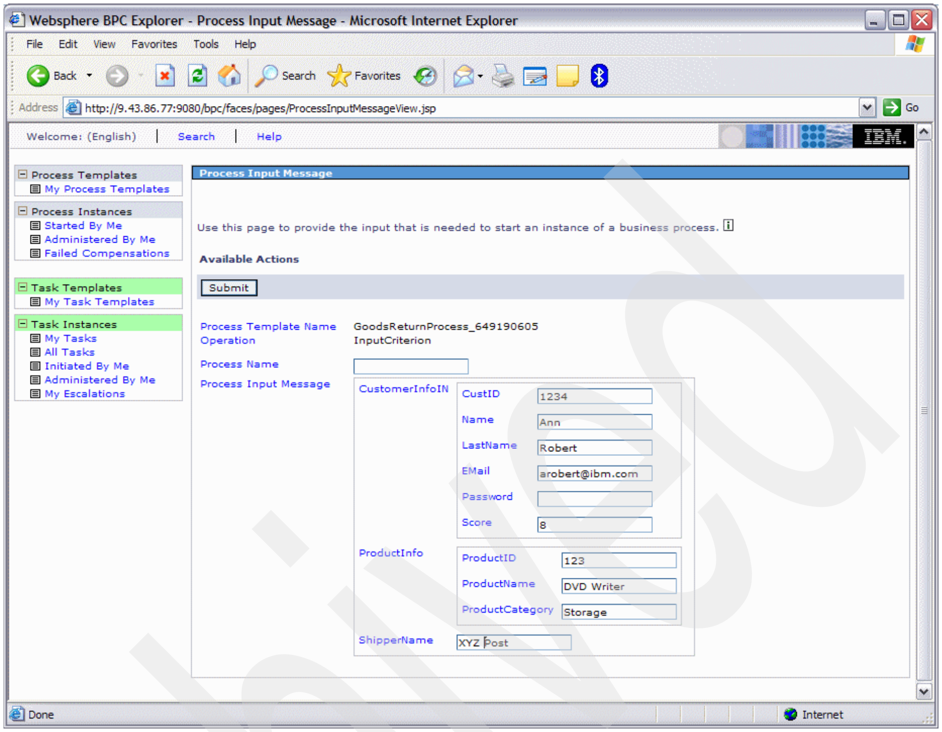


Figure 8-52 Create an instance of GoodsReturnProcess

8. Repeat creating instances of the same product category until you exceed the predefined limit. To create an alert, the number of returns for a specific product category in the current day should exceed the average number of returns per day for this product during the last week.
9. To see the created instances, click again **My Process Templates**, select **GoodsReturnProcess,** and click the **Instances** button.
10. The list of all the instances you have created displays, see Figure 8-53 on page 362.

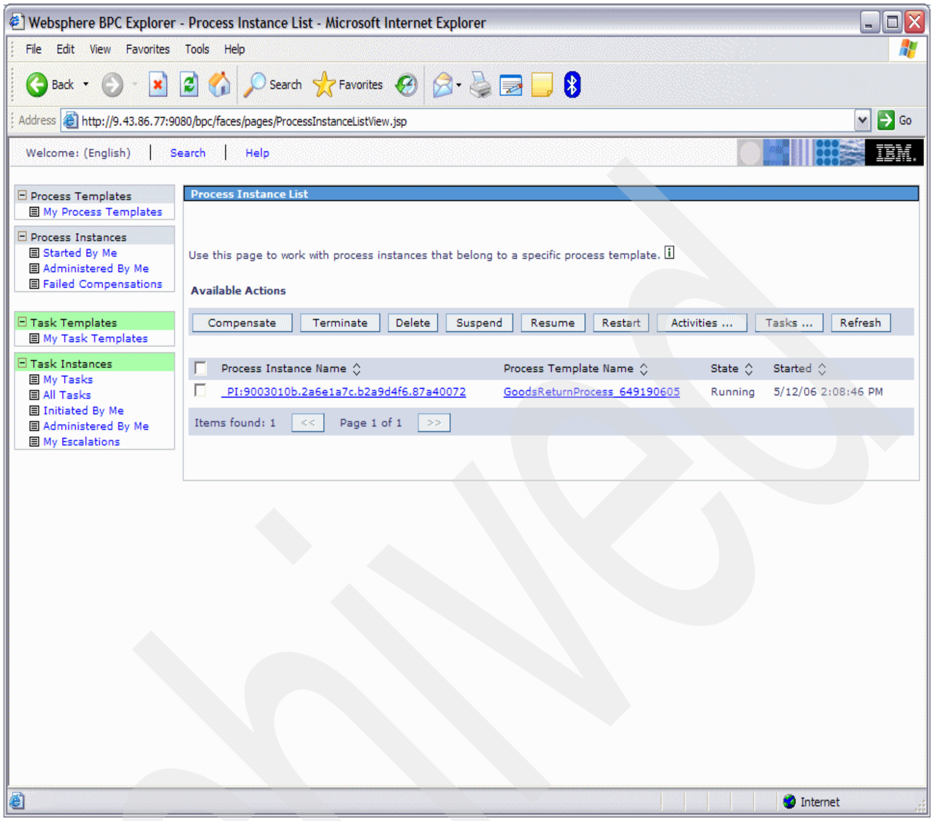


Figure 8-53 An instance created for GoodsReturnProcess

8.7.2 Common Base Event

According to our model, if the number of returns for a certain product category in the current day exceeds the average number of returns per day for this product during the last week, a business situation is detected by WebSphere Business Monitor, and a situation event is thrown. This situation event is called a Common Base Event, and is sent over the CEI (Common Event Infrastructure).

Because CEI is part of WebSphere Process Server, you can see that the event is sent by WebSphere Business Monitor by using an application bundled with WebSphere Process Server called the Common Base Event Browser. To open the Browser:

1. On the WebSphere Business Monitor Server, go to the URL:

   `http://monitor_server_machine:9060/ibm/console`

> **Note:** Because WebSphere Business Monitor Server is responsible for throwing this event on the CEI, the check should be done on the WebSphere Business Monitor Server and not on the Process Server (the event emitter).

2. Log in.
3. From the left pane, expand **Integration Applications**.
4. Click **Common Base Event Browser**. This is depicted in Figure 8-54 on page 364.
5. Click **Get Events** in the new page to get all the events. However, you can also enter a date range to get the events thrown within this range.
6. You will find that number of events has been updated. Click the link **All Events** to show all the events returned.
7. A list of all events is shown. Click the radio button of the event with the latest time stamp. This should be one named **ReturnsAlarmEvent.**
8. Values of the elements in the event are shown. You will find that:
   - **extensionName: ReturnsAlarmEvents:** This is the name of the event template created in WebSphere Business Modeler.
   - **extendedDataElement / BusinessSituationName: HighReturnsRatio:** This is the business situation name previously defined in WebSphere Business Modeler.
   - **extendedDataElement / AlarmValue:12:** This is the number of returns that exceeded the limit.
   - **extendedDataElement / ProductCategory:Storage:** This is the product category that has exceeded the predefined limit in its returns.

   This is the event that will be caught by Adaptive Action Manager, and which subsequently calls the *Transformer* module that, in turn, calls the corrective action process *ProblemEvaluationProcess*.

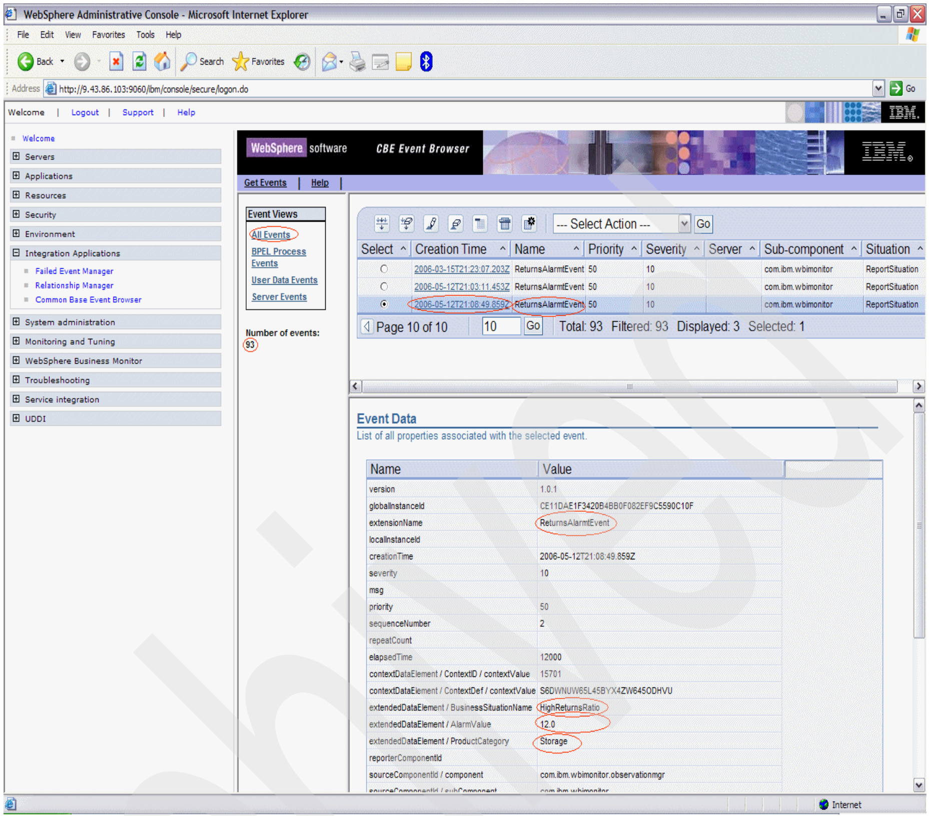


Figure 8-54 Common Base Event Browser

8.7.3 Performing tasks using My Tasks portlet

The situation event that has been thrown by WebSphere Business Monitor Server, has been caught by Adaptive Action Manager. Adaptive Action Manager in turn invokes the *Transformer* module, which is exposed as a Web service. The Transformer module in turn invokes the ProblemEvaluationProcess as a corrective action.

The ProblemEvaluationProcess starts with a human task that is being assigned to the Product Category Manager. The Product Category Manager in turn can see the assigned human tasks using the My Tasks portlet, as explained in section 8.6.1, “My Tasks portlet” on page 326.

When the Product Category Manager displays the dashboards on the Portal server, the My Tasks portlet is showing the newly created human task. Using this portlet, and the other portlets, the Product Category Manager can claim the task and do further root cause analysis. This is explained in section 8.6.1, “My Tasks portlet” on page 326. Based on this information, a decision will be made whether to change the shipper or the vendor. Figure 8-55 depicts a snapshot of the My Tasks portlet, showing the human task we have just created.

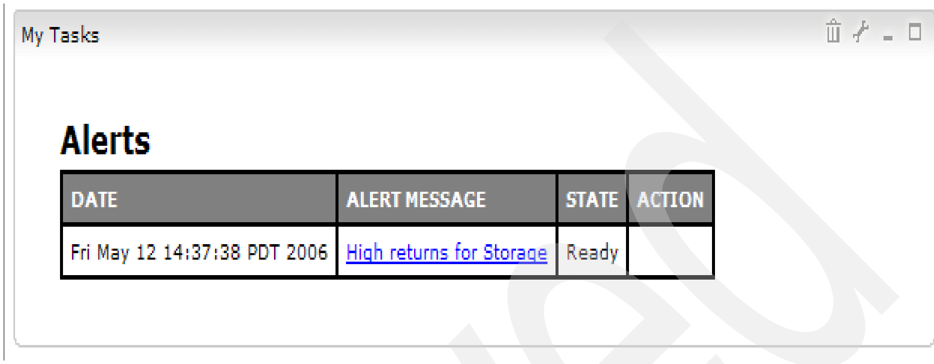


Figure 8-55 My Tasks portlet

A

Portlet implementation code examples

In this appendix, we provide code examples for portal implementation. For example, we include the components of a Human Task portlet and an Alphablox portlet.

Human Task portlet

In this section, we provide Java code examples for the components involved in development of the human task portlet.

Manager component

In Example A-1, we show the Java implementation of the HumanTaskComponent by implementing all of the methods of the HumanTaskInterface, which runs on the WebSphere Process Server.

Example: A-1 HumanTaskManagerComponentImpl.java

```
package com.ibm.websphere.task.sca;

import java.util.List;
import java.util.Vector;

import javax.ejb.CreateException;
import javax.ejb.EJBException;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import com.ibm.task.api.*;
import com.ibm.websphere.bo.BOFactory;
import com.ibm.websphere.sca.ServiceBusinessException;
import com.ibm.websphere.sca.ServiceManager;
import com.ibm.ws.bo.impl.BusObjImpl;

import commonj.sdo.*;


public class HumanTaskManagerComponentImpl {

   private LocalHumanTaskManagerHome taskHome;

   private BOFactory factory;

   /**
    * Default constructor.
    */
   public HumanTaskManagerComponentImpl() {
      super();
   }

   /**
    * Return a reference to the component service instance for this implementation
    * class.  This method should be used when passing this service to a partner reference
```

```
 * or if you want to invoke this component service asynchronously.
 *
 * @generated (com.ibm.wbit.java)
 */
private Object getMyService() {
   return (Object) ServiceManager.INSTANCE.locateService("self");
}

/**
 * Method generated to support implementation of operation "createAndStartTask" defined for
 * WSDL port type named "interface.HumanTaskManager".
 *
 * This WSDL operation has fault(s) defined. Please refer to the WSDL Definition for more
 * information on the type of input, output and fault(s).
 */
public String createAndStartTask(String taskName, String taskNamespace,
      Object inputMessage, Object replyHandlerWrapper)
      throws ServiceBusinessException {
   DataObject result = null;
   DataObject myMessage = null;
   ClientObjectWrapper input = null;
   TaskTemplate[] taskTemplates;
   factory = (BOFactory) new ServiceManager()
         .locateService("com/ibm/websphere/bo/BOFactory");

   try {
      initHTM();
      LocalHumanTaskManager task = taskHome.create();

      try {
         StringBuffer whereClause = new StringBuffer();
         whereClause.append("TASK_TEMPL.NAME = '");
         whereClause.append(taskName);
         whereClause.append("' AND TASK_TEMPL.NAMESPACE = '");
         whereClause.append(taskNamespace);
         whereClause.append("'");
         taskTemplates = task.queryTaskTemplates(whereClause.toString(),
               "TASK_TEMPL.NAME", new Integer(1), null);
         input = task.createInputMessage(taskTemplates[0].getID());
      } catch (NullPointerException npe) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0002E: Task ");
         s.append(taskName);
         s.append(" can not be found deployed on server.  Verify the task name.");
         System.out.println(s.toString());
         npe.printStackTrace();
         throw new ServiceBusinessException(s.toString());
      }
```

```
         if (input.getObject() != null
               && input.getObject() instanceof DataObject) {
            myMessage = (DataObject) input.getObject();
         }

         if (inputMessage != null && inputMessage instanceof DataObject) {
            //assume SCA has converted anyType to DataObject
            Type type1 = myMessage.getType();
            System.out.println("type.getName() = " + type1.getName());
            java.util.List propList1 = type1.getProperties();
            if (propList1.size() == 1) {
               Property prop1 = (Property) propList1.get(0);
               myMessage.set(prop1.getName(), (DataObject) inputMessage);
            } else {
               StringBuffer s = new StringBuffer();
               s.append("HTMF0003E: ");
               s.append("The input message is null or is a primitive part which is not
                        supported at this time.");
               System.out.println(s.toString());
               throw new ServiceBusinessException(s.toString());
            }
            TKIID tkiid = task.createAndStartTask(taskName, taskNamespace, input, null);
            return tkiid.toString();
         } else {
            StringBuffer s = new StringBuffer();
            s.append("HTMF0004E: ");
            s.append("Messages received with primitive types are supported at this time.");
            System.out.println(s.toString());
            throw new ServiceBusinessException(s.toString());
         }
      } catch (TaskException e) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0001E: ");
         s.append("Error occurred within HumanTaskManagerComponent. Check server for more
details.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      } catch (CreateException e) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0005E: ");
         s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      }
   }

   /**
```

```
    * Method generated to support implementation of operation "claimTask" defined for WSDL
port
    * type named "interface.HumanTaskManager".
    *
    * This WSDL operation has fault(s) defined. Please refer to the WSDL Definition for more
    * information on the type of input, output and fault(s).
    */
   public Object claimTask(String tkiid) throws ServiceBusinessException {
      factory = (BOFactory) new ServiceManager()
      .locateService("com/ibm/websphere/bo/BOFactory");

      try {
         initHTM();
         LocalHumanTaskManager task = taskHome.create();
         return (DataObject) task.claim(tkiid).getObject();
      } catch (TaskException e) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0001E: ");
         s.append("Error occurred within HumanTaskManagerComponent.  Check server for more
details.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      } catch (CreateException e) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0005E: ");
         s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      }
   }

   /**
    * Method generated to support implementation of operation "completeTaskWithMessage"
defined
    * for WSDL port type named "interface.HumanTaskManager".
    *
    * This WSDL operation has fault(s) defined. Please refer to the WSDL Definition for more
    * information on the type of input, output and fault(s).
    */
   public void completeTaskWithMessage(String tkiid, Object inputMessage)
         throws ServiceBusinessException {
      factory = (BOFactory) new ServiceManager()
      .locateService("com/ibm/websphere/bo/BOFactory");

      DataObject result = null;
      try {
         initHTM();
         LocalHumanTaskManager task = taskHome.create();
```

```
         ClientObjectWrapper input = task.createOutputMessage(tkiid);

         DataObject myMessage = null;

         if (input.getObject() != null
               && input.getObject() instanceof DataObject) {
            myMessage = (DataObject) input.getObject();

         }
         if (inputMessage != null && inputMessage instanceof DataObject) {
            //assume SCA has converted any to DataObject
            Type type1 = myMessage.getType();
            System.out.println("type.getName() = " + type1.getName());
            java.util.List propList1 = type1.getProperties();
            if (propList1.size() == 1) {
               Property prop1 = (Property) propList1.get(0);
               myMessage.set(prop1.getName(), (DataObject) inputMessage);
            } else {
               StringBuffer s = new StringBuffer();
               s.append("HTMF0003E: ");
               s.append("The input message is null or is a primitive part which is not
supported at this time.");
               System.out.println(s.toString());
               throw new ServiceBusinessException(s.toString());
            }
         }
         task.complete(tkiid, input);
      } catch (TaskException e) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0001E: ");
         s.append("Error occurred within HumanTaskManagerComponent.  Check server for more
details.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      } catch (CreateException e) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0005E: ");
         s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      }

   }

   /**
    * Method generated to support implementation of operation "getTaskByID"
    * defined for WSDL port type named "interface.HumanTaskManager".
    *
```

```
     * This WSDL operation has fault(s) defined. The presence of
     * commonj.sdo.DataObject as the return type and/or as a parameter type
     * conveys that its a complex type. Please refer to the WSDL Definition for
     * more information on the type of input, output and fault(s).
     */
    public DataObject getTaskByID(String tkiid) throws ServiceBusinessException {
       System.out.println("hello");
       DataObject result = null;
       boolean taskfound = false;
       factory = (BOFactory) new
ServiceManager().locateService("com/ibm/websphere/bo/BOFactory");

       try {
          initHTM();
          LocalHumanTaskManager task = taskHome.create();

          QueryResultSet[] resultSetArray = new QueryResultSet[1];
          String selectClause = "DISTINCT
TASK.ACTIVATED,TASK.COMPLETED,TASK.DUE,TASK.EXPIRES,TASK.FIRST_ACTIVATED,TASK.KIND,TASK.LAST_MO
DIFIED,TASK.LAST_STATE_CHANGE, TASK.NAME,TASK.NAME_SPACE,TASK.ORIGINATOR, TASK.OWNER,
TASK.PRIORITY, TASK.STARTER, TASK.STARTED,
TASK.STATE,TASK.TYPE,TASK.IS_ESCALATED,TASK.IS_INLINE,TASK.SUSPENDED,TASK.SUPPORT_AUTOCLAIM,TAS
K.SUPPORT_CLAIM_SUSP,TASK.SUPPORT_DELEGATION, TASK.TKIID";
          String whereClause = "TASK.TKIID = ID('" + tkiid + "')";

          resultSetArray[0] = task
                .query(selectClause, whereClause, null,new Integer(1), null);

          while (resultSetArray[0].next()) {
             taskfound = true;
             result = buildTask(resultSetArray);
          }

          if (!taskfound) {
             StringBuffer s = new StringBuffer();
             s.append("HTMF0006E: ");
             s.append("Task not Found for ID ");
             s.append(tkiid);
             System.out.println(s.toString());
             throw new ServiceBusinessException(s.toString());
          }

          return result;

       } catch (TaskException e) {
          StringBuffer s = new StringBuffer();
          s.append("HTMF0001E: ");
          s.append("Error occurred within HumanTaskManagerComponent.  Check server for more
details.");
```

```
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      } catch (CreateException e) {
         StringBuffer s = new StringBuffer();
         s.append("HTMF0005E: ");
         s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      }

   }

   /**
    * Method generated to support implementation of operation "getTasks" defined
    * for WSDL port type named "interface.HumanTaskManager".
    *
    * This WSDL operation has fault(s) defined. The presence of
    * commonj.sdo.DataObject as the return type and/or as a parameter type
    * conveys that its a complex type. Please refer to the WSDL Definition for
    * more information on the type of input, output and fault(s).
    */
   public DataObject getTasks(String whereClause, String orderBy,
         Integer skipTuples, Integer threshold)
         throws ServiceBusinessException {
      System.out.println("Test");
      DataObject result, output = null;
      boolean taskfound = false;
      List list = new Vector();
      factory = (BOFactory) new ServiceManager()
      .locateService("com/ibm/websphere/bo/BOFactory");

      try {
         initHTM();
         LocalHumanTaskManager task = taskHome.create();
         QueryResultSet[] resultSetArray = new QueryResultSet[1];
         resultSetArray[0] = task
               .query(
                     "DISTINCT
TASK.ACTIVATED,TASK.COMPLETED,TASK.DUE,TASK.EXPIRES,TASK.FIRST_ACTIVATED,TASK.KIND,TASK.LAST_MO
DIFIED,TASK.LAST_STATE_CHANGE, TASK.NAME,TASK.NAME_SPACE,TASK.ORIGINATOR, TASK.OWNER,
TASK.PRIORITY, TASK.STARTER, TASK.STARTED,
TASK.STATE,TASK.TYPE,TASK.IS_ESCALATED,TASK.IS_INLINE,TASK.SUSPENDED,TASK.SUPPORT_AUTOCLAIM,TAS
K.SUPPORT_CLAIM_SUSP,TASK.SUPPORT_DELEGATION,TASK.TKIID",
                     whereClause, orderBy, skipTuples, threshold, null);


         while (resultSetArray[0].next()) //0 to n
         {
```

```
            taskfound = true;
            result = buildTask(resultSetArray);
            list.add(result);
            System.out.println("Get Properties");
            try {
               if(resultSetArray[0].getInteger(6).intValue()==105) {
                  System.out.println("get Input:");
                  try {

                  ClientObjectWrapper clientObjectWrapper =
task.getInputMessage((resultSetArray[0].getOID(24)).toString());
                  if(clientObjectWrapper!=null) {
                     System.out.println("Got ClientObjectWrapper");
                     DataObject dataObject = (DataObject)clientObjectWrapper.getObject();
                     if(dataObject!=null) {
                        System.out.println("Got
DataObject:"+dataObject+":"+dataObject.getType());
                        //BusObjImpl busObjImpl = (BusObjImpl)dataObject.get(0);
                        System.out.println("Got BusObj");
                        Type type = dataObject.getType();
                        System.out.println("Got Type");
                        List propList = type.getProperties();
                        System.out.println("Got list");
                        for(int i=0;i<propList.size();i++) {
                           Property property = (Property) propList.get(i);
                           System.out.println("Gor Prop:"+property);
                           System.out.println("input:"+property.getName()+"=");
                        }
                     }
                  }
                  } catch(Exception e) {
                     System.out.println("Input Exception:"+e.getMessage());
                  }
                  try {
                  System.out.println("Get Output");
                  ClientObjectWrapper clientObjectWrapper1 =
task.getOutputMessage((resultSetArray[0].getOID(24)).toString());
                  if(clientObjectWrapper1!=null) {
                     DataObject dataObject = (DataObject)clientObjectWrapper1.getObject();
                     if(dataObject!=null) {
                        BusObjImpl busObjImpl = (BusObjImpl)dataObject.get(0);
                        Type type = busObjImpl.getType();
                        List propList = type.getProperties();
                        for(int i=0;i<propList.size();i++) {
                           Property property = (Property) propList.get(i);
                           System.out.println("Output:"+property.getName()+"=");
                        }
                     }
                  }
```

```
                    }catch(Exception e) {
                       System.out.println("Export Exception:"+e.getMessage());
                    }
                 }

              } catch (EJBException e1) {
                 // XXX Auto-generated catch block
                 e1.printStackTrace();
              }
           }

           if (!taskfound) {
              String s = "HTMF0007E: No tasks found";
              System.out.println(s);
              throw new ServiceBusinessException(s);
           }

           output = factory.create(
                    "http://BFMHTMLibrary/com/ibm/websphere/htm/queryresulttask",
                    "QueryResultTask");
           output.setList("tasks", list);
           return output;

        } catch (TaskException e) {
           StringBuffer s = new StringBuffer();
           s.append("HTMF0001E: ");
           s.append("Error occurred within HumanTaskManagerComponent.  Check server for more
details.");
           System.out.println(s.toString());
           throw new ServiceBusinessException(s.toString());
        } catch (CreateException e) {
           StringBuffer s = new StringBuffer();
           s.append("HTMF0005E: ");
           s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
           System.out.println(s.toString());
           throw new ServiceBusinessException(s.toString());
        }
     }

     /**
      * Method generated to support implementation of operation "getTaskIDs" defined for WSDL
port
      * type named "interface.HumanTaskManager".
      *
      * This WSDL operation has fault(s) defined. The presence of commonj.sdo.DataObject as the
      * return type and/or as a parameter type conveys that its a complex type. Please refer to
      * the WSDL Definition for more information on the type of input, output and fault(s).
      */
```

```
public DataObject getTaskIDs(String whereClause, String orderBy,
      Integer skipTuples, Integer threshold)
      throws ServiceBusinessException {
   DataObject tkiid, result, output = null;
   boolean taskfound = false;
   List list = new Vector();
   factory = (BOFactory) new ServiceManager()
   .locateService("com/ibm/websphere/bo/BOFactory");

   try {
      initHTM();
      LocalHumanTaskManager task = taskHome.create();

      QueryResultSet resultSet = null;

      resultSet = task.query("DISTINCT TASK.TKIID", whereClause, orderBy,
            skipTuples, threshold, null);

      while (resultSet.next()) //0 to n
      {
         list.add(resultSet.getOID(1).toString());
         taskfound = true;
      }

      if (!taskfound) {
         String s = "HTMF0007E: No tasks found";
         System.out.println(s);
         throw new ServiceBusinessException(s);
      }
      output = factory
      .create(
            "http://BFMHTMLibrary/com/ibm/websphere/htm/queryresulttaskid",
            "QueryResultTaskID");
output.setList("taskIDs", list);
return output;

   } catch (TaskException e) {
      e.printStackTrace();
      String s = "HTMF0002E: Task Exception";
      System.out.println(s);
      result = factory.create(
            "http://BFMHTMLibrary/com/ibm/websphere/htm/sca/bofault",
            "FaultTask");
      result.setString("faultID", "HTMF002E");
      result.setString("faultValue", "Task Exception");
      throw new ServiceBusinessException(result);
   }  catch (CreateException e) {
      StringBuffer s = new StringBuffer();
      s.append("HTMF0005E: ");
```

```
         s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
         System.out.println(s.toString());
         throw new ServiceBusinessException(s.toString());
      }
   }

   private DataObject buildTask(QueryResultSet resultSet[]) {
      DataObject result = factory.create(
            "http://BFMHTMLibrary/com/ibm/websphere/htm/task", "Task");

      try {
         result.setDate("activationTime", resultSet[0].getTimestamp(1)
               .getTime());
      } catch (NullPointerException npe) {
      }
      try {
         result.setDate("completionTime", resultSet[0].getTimestamp(2)
               .getTime());
      } catch (NullPointerException npe) {
      }
      try {
         result.setDate("dueTime", resultSet[0].getTimestamp(3).getTime());
      } catch (NullPointerException npe) {
      }
      try {
         result.setDate("expirationTime", resultSet[0].getTimestamp(4)
               .getTime());
      } catch (NullPointerException npe) {
      }
      try {
         result.setDate("firstActivationTime", resultSet[0].getTimestamp(5)
               .getTime());
      } catch (NullPointerException npe) {

      }
      try {
         result.setInt("kind", resultSet[0].getInteger(6).intValue());
      } catch (NullPointerException npe) {
      }
      try {
         result.setDate("lastModificationTime", resultSet[0].getTimestamp(7)
               .getTime());
      } catch (NullPointerException npe) {
      }
      try {
         result.setDate("lastStateChangeTime", resultSet[0].getTimestamp(8)
               .getTime());
      } catch (NullPointerException npe) {
```

```
}
try {
   result.setString("name", resultSet[0].getString(9));
} catch (NullPointerException npe) {
}
try {
   result.setString("namespace", resultSet[0].getString(10));
} catch (NullPointerException npe) {
}
try {
   result.setString("originator", resultSet[0].getString(11));
} catch (NullPointerException npe) {
}
try {
   result.setString("owner", resultSet[0].getString(12));
} catch (NullPointerException npe) {
}
try {
   result.setInt("priority", resultSet[0].getInteger(13).intValue());
} catch (NullPointerException npe) {
}
try {
   result.setString("starter", resultSet[0].getString(14));
} catch (NullPointerException npe) {
}
try {
   result
         .setDate("startTime", resultSet[0].getTimestamp(15)
               .getTime());
} catch (NullPointerException npe) {
}
try {
   result.setInt("state", resultSet[0].getInteger(16).intValue());
} catch (NullPointerException npe) {
}
try {
   result.setString("type", resultSet[0].getString(17));
} catch (NullPointerException npe) {
}
try {
   result.setBoolean("escalated", resultSet[0].getBoolean(18)
         .booleanValue());
} catch (NullPointerException npe) {
}
try {
   result.setBoolean("inline", resultSet[0].getBoolean(19)
         .booleanValue());
} catch (NullPointerException npe) {
}
```

```
      try {
         result.setBoolean("suspended", resultSet[0].getBoolean(20)
               .booleanValue());
      } catch (NullPointerException npe) {
      }
      try {
         result.setBoolean("autoclaim", resultSet[0].getBoolean(21)
               .booleanValue());
      } catch (NullPointerException npe) {
      }
      try {
         result.setBoolean("claimsuspend", resultSet[0].getBoolean(22)
               .booleanValue());
      } catch (NullPointerException npe) {
      }
      try {
         result.setBoolean("delegation", resultSet[0].getBoolean(23)
               .booleanValue());
      } catch (NullPointerException npe) {
      }
      try {
         result.setString("tkiid", (resultSet[0].getOID(24)).toString());
      } catch (NullPointerException npe) {
      }

      return result;
   }

   private void initHTM()
   {
      try
      {
      InitialContext initialContext = new InitialContext();

      // Lookup the local home interface of the LocalHumanTaskManager bean

      taskHome = (LocalHumanTaskManagerHome) initialContext
            .lookup("java:comp/env/ejb/LocalHumanTaskManagerHome");

      }
   catch (NamingException e) {
      System.out.println("Lookup for Human Task Manager local interface (EJB) failed");
      e.printStackTrace();
      throw new ServiceBusinessException("facade null");

   }

   }
```

```
    /**
     * Method generated to support implementation of operation "getInputMessage" defined for
WSDL
     * port type named "interface.HumanTaskManager".
     *
     * This WSDL operation has fault(s) defined. Please refer to the WSDL Definition for more
     * information on the type of input, output and fault(s).
     */
    public String getInputMessage(String tkiid, String property) throws
ServiceBusinessException {
        factory = (BOFactory) new
ServiceManager().locateService("com/ibm/websphere/bo/BOFactory");

        try {
            initHTM();
            LocalHumanTaskManager task = taskHome.create();
            ClientObjectWrapper clientObjectWrapper = task.getInputMessage(tkiid);
            String inputMessage = null;
            if(clientObjectWrapper!=null) {
                DataObject dataObject = (DataObject)clientObjectWrapper.getObject();
                if(dataObject!=null) {
                    Type type = dataObject.getType();
                    inputMessage = dataObject.getString(property);
                }
            }
            return inputMessage;

        } catch (TaskException e) {
            StringBuffer s = new StringBuffer();
            s.append("HTMF0001E: ");
            s.append("Error occurred within HumanTaskManagerComponent.  Check server for more
details.");
            System.out.println(s.toString());
            throw new ServiceBusinessException(s.toString());
        } catch (CreateException e) {
            StringBuffer s = new StringBuffer();
            s.append("HTMF0005E: ");
            s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
            System.out.println(s.toString());
            throw new ServiceBusinessException(s.toString());
        }

    }

    /**
     * Method generated to support implementation of operation "getOutputMessage" defined for
     * WSDL port type named "interface.HumanTaskManager".
     *
```

```
     * This WSDL operation has fault(s) defined. Please refer to the WSDL Definition for more
     * information on the type of input, output and fault(s).
     */
    public String getOutputMessage(String tkiid, String property)
          throws ServiceBusinessException {
       factory = (BOFactory) new
ServiceManager().locateService("com/ibm/websphere/bo/BOFactory");

       try {
          initHTM();
          LocalHumanTaskManager task = taskHome.create();
          ClientObjectWrapper clientObjectWrapper = task.getOutputMessage(tkiid);
          String inputMessage = null;
          if(clientObjectWrapper!=null) {
             DataObject dataObject = (DataObject)clientObjectWrapper.getObject();
             if(dataObject!=null) {
                Type type = dataObject.getType();
                inputMessage = dataObject.getString(property);
             }
          }
          return inputMessage;

       } catch (TaskException e) {
          StringBuffer s = new StringBuffer();
          s.append("HTMF0001E: ");
          s.append("Error occurred within HumanTaskManagerComponent.  Check server for more
details.");
          System.out.println(s.toString());
          throw new ServiceBusinessException(s.toString());
       } catch (CreateException e) {
          StringBuffer s = new StringBuffer();
          s.append("HTMF0005E: ");
          s.append("Error occurred with retrieving HumanTaskManager in
HumanTaskManagerComponent.  Check availability of HumanTaskManager.");
          System.out.println(s.toString());
          throw new ServiceBusinessException(s.toString());
       }
    }
}
```

Portlet view mode

In this section, we show the jsp code for the view mode of the Human Task portlet. It is also referred to as the My Tasks portlet. The jsp code is depicted in Example A-2 on page 383.

Example: A-2 HumanTaskPortletView.jsp

```
<%@ page session="false" contentType="text/html" import="java.util.*,
humantask.*,org.apache.jetspeed.portlet.*"%>
<%@ taglib uri="/WEB-INF/tld/c2a.tld" prefix="C2A" %>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init/>

<%
      // Parameters for edit mode
   PortletData portletData = portletRequest.getData();
   // Parameters for config mode
   PortletSettings portletSettings = portletRequest.getPortletSettings();

   HumanTaskPortletSessionBean sessionBean =
      (HumanTaskPortletSessionBean)portletRequest.
         getPortletSession().getAttribute(HumanTaskPortlet.SESSION_BEAN);
%>

<DIV style="margin: 6px">

<H3 style="margin-bottom: 3px">Alerts</H3>

<%
   BFMHTMLibrary.HumanTaskManagerProxy sampleHumanTaskManagerProxyid =
      new BFMHTMLibrary.HumanTaskManagerProxy(portletSettings.getAttribute("url"));

request.getSession().setAttribute("sampleHumanTaskManagerProxyid",sampleHumanTaskManagerProxyid
);
%>
<table cellpadding="5" cellspacing="0" style="border:1px solid black">
   <tr>
      <td style="border:1px solid
black;background-color:gray;color:white;"><STRONG>DATE</STRONG></td>
      <td style="border:1px solid black;background-color:gray;color:white;"><STRONG>ALERT
MESSAGE</STRONG></td>
      <td style="border:1px solid
black;background-color:gray;color:white;"><STRONG>STATE</STRONG></td>
      <td style="border:1px solid
black;background-color:gray;color:white;"><STRONG>ACTION</STRONG></td>
   </tr>
<%
try {
    String whereClause = portletSettings.getAttribute("whereClause");
    String orderBy = portletSettings.getAttribute("orderBy");
    String skipTuples = portletSettings.getAttribute("skipTuples");
    if(skipTuples==null)
    skipTuples = "0";
    Integer skipTuplesInteger  = Integer.valueOf(skipTuples);
```

```
    String threshold =  portletSettings.getAttribute("threshold");
    if(threshold==null)
    threshold = "0";
    Integer thresholdInteger  = Integer.valueOf(threshold);
    BFMHTMLibrary.QueryResultTask getTasks =
sampleHumanTaskManagerProxyid.getTasks(whereClause,orderBy,skipTuplesInteger,thresholdInteger);
   if(getTasks!= null){
      BFMHTMLibrary.Task[] tasks = getTasks.getTasks();
      for(int i=0;i<tasks.length;i++) {
         BFMHTMLibrary.Task task = tasks[i];
         String inputMessage = null;
         if(task.getKind().intValue()==105) {
            inputMessage =
sampleHumanTaskManagerProxyid.getInputMessage(task.getTkiid(),"ProductCategoryAPIn");
%>
<tr>
   <td style="border:1px solid black"><%=task.getActivationTime().getTime() %></td>
   <td style="border:1px solid black"><a href="
      <portletAPI:createURI>
         <portletAPI:URIAction name='<%=HumanTaskPortlet.FORM_ACTION%>'/>
         <portletAPI:URIParameter name="inputMessage" value="<%=inputMessage %>"/>
         <portletAPI:URIParameter name="tkiid" value="<%=task.getTkiid()%>"/>
      </portletAPI:createURI>
      ">High returns for <%=inputMessage %></a>
   </td>
   <td style="border:1px solid black"><%=getStateName(task.getState()) %></td>
   <td style="border:1px solid black"> 
      <%
         if(task.getState().intValue()==1000) {
      %>
      <a href="
      <portletAPI:createURI>
         <portletAPI:URIAction name="<%=HumanTaskPortlet.TASK_COMPLETE_ACTION%>"/>
         <portletAPI:URIParameter name="tkiid" value="<%=task.getTkiid()%>"/>
      </portletAPI:createURI>
      ">Complete</a>
      <%
         }
      %>
   </td>
</tr>
<%
            }
      }
    }

} catch (Exception e) {
%>
exception: <%= e %>
```

```
<%
return;
}
%>
</table>
<% /******** End of sample code *********/ %>

<%
String message = sessionBean.getValue(HumanTaskPortlet.MESSAGE);
String tkiid = sessionBean.getValue(HumanTaskPortlet.CURRENT_TKIID);
if(tkiid!=null && message!=null && message.startsWith(HumanTaskPortlet.CHANGE_SHIPPER_PREFIX))
{
   String shipper =
message.substring(message.indexOf(HumanTaskPortlet.CHANGE_SHIPPER_PREFIX)+HumanTaskPortlet.CHAN
GE_SHIPPER_PREFIX.length());
%>
<h3>Change Shipper</h3>
Please enter the new shipper.
<FORM method="POST" action="<portletAPI:createURI><portletAPI:URIAction
name='<%=HumanTaskPortlet.TASK_COMPLETE_ACTION%>'/></portletAPI:createURI>">
    <INPUT class="wpsEditField"  name="<%=HumanTaskPortlet.TKIID %>" value="<%=tkiid%>"
type="hidden"/>
    <LABEL class="wpsLabelText"   for="currentShipper">Current Shipper: </LABEL>
    <INPUT class="wpsEditField"  name="currentShipper" value="<%=shipper%>" type="text"/>
       
    <LABEL class="wpsLabelText"   for="newShipper">New Shipper</LABEL>
    <INPUT class="wpsEditField"  name="newShipper" value="" type="text"/><BR>
    <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=HumanTaskPortlet.SUBMIT%>'/>" value="Complete" type="submit"/>
</FORM>
<%
}
sessionBean.setValue(HumanTaskPortlet.MESSAGE,null);
 %>
</DIV>
<%!
   public String getStateName(Integer state) {
      if(2==state.intValue())
         return "Ready";
      if(8==state.intValue())
         return "Claimed";
      return state.toString();
   }
%>
```

Configuration mode

In this section, we show the jsp code for the configuration mode of the Human Task portlet, also referred to as the My Tasks portlet. This is the portlet that enables users to change the query parameters for the human tasks. The code for that portlet is depicted in Example A-3.

Example: A-3 HumanTaskPortletConfig.jsp

```
<%@ page session="false" contentType="text/html"
import="org.apache.jetspeed.portlet.*,humantask.*" %>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init/>

<DIV style="margin: 6px">

<H3 style="margin-bottom: 3px">Welcome!</H3>
<% /******** Start of sample code ********/ %>
  <%
  PortletSettings portletSettings = portletRequest.getPortletSettings();
  if( portletSettings!=null ) {
   String url = portletSettings.getAttribute("url");
   if(url==null)
      url = "http://9.43.86.77:9080/BFMHTMFacadeWeb/sca/HumanTaskManagerComponentExport";
    String whereClause = portletSettings.getAttribute("whereClause");
    String orderBy = portletSettings.getAttribute("orderBy");
    String skipTuples = portletSettings.getAttribute("skipTuples");
    if(skipTuples==null)
    skipTuples = "0";
    String threshold =  portletSettings.getAttribute("threshold");
    if(threshold==null)
    threshold = "0";

   %>
  <FORM method="POST" action="<portletAPI:createURI><portletAPI:URIAction
name='<%=HumanTaskPortlet.CONFIG_ACTION%>'/></portletAPI:createURI>">
    <LABEL class="wpsLabelText"   for="url">URL</LABEL><BR>
    <INPUT class="wpsEditField"  name="url" value="<%=getParam(url)%>" type="text"/><BR>
    <LABEL class="wpsLabelText"   for="whereClause">Where Clause</LABEL><BR>
    <INPUT class="wpsEditField"  name="whereClause" value="<%=getParam(whereClause)%>"
type="text"/><BR>
    <LABEL class="wpsLabelText"   for="orderBy">Order By</LABEL><BR>
    <INPUT class="wpsEditField"  name="orderBy" value="<%=getParam(orderBy)%>"
type="text"/><BR>
    <LABEL class="wpsLabelText"   for="skipTuples">Skip Tuples</LABEL><BR>
    <INPUT class="wpsEditField"  name="skipTuples" value="<%=getParam(skipTuples)%>"
type="text"/><BR>
    <LABEL class="wpsLabelText"   for="threshold">Threshold</LABEL><BR>
```

```
    <INPUT class="wpsEditField"  name="threshold" value="<%=getParam(threshold)%>"
type="text"/><BR>
    <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=HumanTaskPortlet.SUBMIT%>'/>" value="Save" type="submit"/>
    <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=HumanTaskPortlet.CANCEL%>'/>" value="Cancel" type="submit"/>
  </FORM>
<% /******** End of sample code *********/ %>
</DIV>
<%
}
 %>
<%!
public Object getParam(Object value) {
   if(value==null)
      return "";
   else if(value.equals("null"))
      return "";
   return value;
}
%>
```

Human Task session bean

In this section, we show the java code for the session bean of the Human Task portlet, also refered to as the My Tasks portlet. This session bean stores name value pairs in the session object. The code is depicted in Example A-4.

Example: A-4   HumanTaskPortletSessionBean.java

```
package humantask;

import java.util.HashMap;

/**
 *
 * A sample Java bean that stores portlet instance data in portlet session.
 *
 */
public class HumanTaskPortletSessionBean {

   //**********************************************************
   //* Last text for the text form
   //**********************************************************

   private HashMap hashMap = new HashMap();

   /**
```

```
     * Set last text for the text form.
     *
     * @param formText last text for the text form.
     */
    public void setValue(String param,String value) {
        this.hashMap.put(param,value);
    }

   /**
     * Get last text for the text form.
     *
     * @return last text for the text form
     */
    public String getValue(String param) {
        return (String)this.hashMap.get(param);
    }

}
```

Portlet controller

In this section, we show the java code for the portlet controller of the Human Task portlet, also refered to as My Tasks portlet. This code controls the actions of the portlet, such as *claim task* and *complete task*. The code is depicted in Example A-5.

Example: A-5 HumanTaskPortlet.java

```
package humantask;

import java.io.IOException;
import java.io.PrintWriter;
import java.io.Writer;
import java.rmi.RemoteException;
import java.util.Enumeration;

import org.apache.jetspeed.portlet.*;
import org.apache.jetspeed.portlet.event.*;

import BFMHTMLibrary.ClaimTask_faultMessageMsg;
import BFMHTMLibrary.CompleteTaskWithMessage_faultMessageMsg;

/**
 *
 * A sample portlet based on PortletAdapter
 *
 */
```

```
public class HumanTaskPortlet extends PortletAdapter implements ActionListener,MessageListener
{

    public static final String VIEW_JSP         = "/humantask/jsp/HumanTaskPortletView.";
// JSP file name to be rendered on the view mode
    public static final String CONFIG_JSP       = "/humantask/jsp/HumanTaskPortletConfig.";
// JSP file name to be rendered on the configure mode
    public static final String SESSION_BEAN     = "humantask.HumanTaskPortletSessionBean";
// Bean name for the portlet session
    public static final String FORM_ACTION      = "humantask.HumanTaskPortletFormAction";
// Action name for the orderId entry form
    public static final String TEXT             = "humantask.HumanTaskPortletText";
// Parameter name for general text input
    public static final String SUBMIT           = "humantask.HumanTaskPortletSubmit";
// Parameter name for general submit button
    public static final String CANCEL           = "humantask.HumanTaskPortletCancel";
// Parameter name for general cancel button

    public static final String CONFIG_ACTION    = "humantask.HumanTaskPortletConfigAction";
// Action name for the configure form
    public static final String CONFIG_NAME      = "humantask.HumanTaskPortletConfigName";
// Attribute name for the PortletSettings
   public static final String SOURCE       = "HumanTaskPortlet_orderId";
// Parameter name for cooperative source

    public static final String CURRENT_TKIID= "humantask.CurrentTkiid";
    public static final String TKIID = "tkiid";

    public static final String MESSAGE= "humantask.message";
    public static final String TASK_COMPLETE_ACTION= "humantask.TaskCompleteAction";
    public static final String CHANGE_SHIPPER_PREFIX = "ChangeShipper:";
    public static final String INPUT_MESSAGE= "inputMessage";
    /**
     * @see org.apache.jetspeed.portlet.Portlet#init(PortletConfig)
     */
   public void init(PortletConfig portletConfig) throws UnavailableException {
      super.init(portletConfig);
   }

   /**
    * @see org.apache.jetspeed.portlet.PortletAdapter#doView(PortletRequest, PortletResponse)
    */
   public void doView(PortletRequest request, PortletResponse response) throws
PortletException, IOException {
      // Check if portlet session exists
      HumanTaskPortletSessionBean sessionBean = getSessionBean(request);
      try {
      if( sessionBean==null ) {
          response.getWriter().println("<b>NO PORTLET SESSION YET</b>");
```

```
            return;
         }

          // Invoke the JSP to render
         getPortletConfig().getContext().include(VIEW_JSP+getJspExtension(request), request,
response);
      } catch (Throwable exc) {
            Writer writer = response.getWriter();
            writer.write("<pre>");
            exc.printStackTrace(new PrintWriter(writer));
            writer.write("</pre>");
        }
   }

   /**
    * @see org.apache.jetspeed.portlet.PortletAdapter#doConfigure(PortletRequest,
PortletResponse)
    */
   public void doConfigure(PortletRequest request, PortletResponse response) throws
PortletException, IOException {
      try {
         // Invoke the JSP to render
         getPortletConfig().getContext().include(CONFIG_JSP+getJspExtension(request), request,
response);
      } catch (Throwable exc) {
            Writer writer = response.getWriter();
            writer.write("<pre>");
            exc.printStackTrace(new PrintWriter(writer));
            writer.write("</pre>");
        }
   }

   /**
    * @see org.apache.jetspeed.portlet.event.ActionListener#actionPerformed(ActionEvent)
    */
   public void actionPerformed(ActionEvent event) throws PortletException {
      if( getPortletLog().isDebugEnabled() )
         getPortletLog().debug("ActionListener - actionPerformed called");
      // ActionEvent handler
      String actionString = event.getActionString();
      PortletRequest request = event.getRequest();
      // Add action string handler here
      HumanTaskPortletSessionBean sessionBean = getSessionBean(request);
        if( FORM_ACTION.equals(actionString) ) {
         String inputMessage = request.getParameter("inputMessage");
         String tkiid = request.getParameter(TKIID);
         BFMHTMLibrary.HumanTaskManagerProxy sampleHumanTaskManagerProxyid =
(BFMHTMLibrary.HumanTaskManagerProxy)request.getSession().getAttribute("sampleHumanTaskManagerP
roxyid");
```

```
         try {
            System.out.println("Claim:"+tkiid);
            sessionBean.setValue(CURRENT_TKIID,tkiid);
            sampleHumanTaskManagerProxyid.claimTask(tkiid);

         } catch (RemoteException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
         } catch (ClaimTask_faultMessageMsg e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
         }
         PortletMessage message = new DefaultPortletMessage(inputMessage);
           getPortletConfig().getContext().send(null,message);
      }
        if( TASK_COMPLETE_ACTION.equals(actionString) ) {
         String tkiid = request.getParameter(TKIID);
         BFMHTMLibrary.HumanTaskManagerProxy sampleHumanTaskManagerProxyid =
(BFMHTMLibrary.HumanTaskManagerProxy)request.getSession().getAttribute("sampleHumanTaskManagerP
roxyid");
         try {
            System.out.println("Complete:"+tkiid);
            sessionBean.setValue(MESSAGE,null);
            sessionBean.setValue(CURRENT_TKIID,null);
            sampleHumanTaskManagerProxyid.completeTaskWithMessage(tkiid,null);
            // Set session variables to null
         } catch (RemoteException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
         } catch (CompleteTaskWithMessage_faultMessageMsg e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
         }
      }
      if( CONFIG_ACTION.equals(actionString) ) {
         if( request.getParameter(SUBMIT)!=null ) {
            PortletSettings settings = request.getPortletSettings();
            if( settings!=null ) {
//               --> Loop through parameters
               Enumeration enum = request.getParameterNames();
               while(enum.hasMoreElements()) {
                  String key = (String)enum.nextElement();
                  String value = request.getParameter(key);
                  if(key!=null && value!=null && !value.equals("null"))
                     settings.setAttribute(key,value);
               }
               try {
                 settings.store();// save to data store
               }
```

```
                catch (IOException ioe) {
                   if( getPortletLog().isErrorEnabled() )
                      getPortletLog().error("Error on PortletSettings.store():
"+ioe.getMessage());
                }
             }
          }
       }
   }

   /**
    * Get SessionBean.
    *
    * @param request PortletRequest
    * @return HumanTaskPortletSessionBean
    */
   private HumanTaskPortletSessionBean getSessionBean(PortletRequest request) {
        PortletSession session = request.getPortletSession(false);
        if( session == null )
        return null;
        HumanTaskPortletSessionBean sessionBean =
(HumanTaskPortletSessionBean)session.getAttribute(SESSION_BEAN);
        if( sessionBean == null ) {
        sessionBean = new HumanTaskPortletSessionBean();
        session.setAttribute(SESSION_BEAN,sessionBean);
        }
        return sessionBean;
   }

   public void messageReceived(MessageEvent event) {
      System.out.println("Got message");
      PortletMessage message = event.getMessage();
      if (message instanceof DefaultPortletMessage) {
          String messageString = ((DefaultPortletMessage)message).getMessage();
          HumanTaskPortletSessionBean sessionBean = getSessionBean(event.getRequest());
          sessionBean.setValue(MESSAGE,messageString);
       }
   }

   /**
    * Returns the file extension for the JSP file
    *
    * @param request PortletRequest
    * @return JSP extension
    */
   private static String getJspExtension(PortletRequest request) {
      String markupName = request.getClient().getMarkupName();
      return "jsp";
   }
```

```
}
```

Alphablox portlet

In this section, are examples that illustrate the code necessary to build the Alphablox portlet.

Portlet View Mode

In this section we show the jsp code for the view mode of the Alphablox Portlet. The code is depicted in Example A-6.

Example: A-6 AlphabloxPortletView.jsp

```
<%@ page import="com.alphablox.blox.data.mdb.*,
             com.alphablox.blox.uimodel.*,
                com.alphablox.blox.repository.*,
                org.apache.jetspeed.portlet.*,
                alphablox.*" %>

<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="bloxuitld" prefix="bloxui"%>
<%@ taglib uri="bloxportlettld" prefix="bloxportlet"%>


<portletAPI:init/>

<blox:header/>

<%
   AlphabloxPortletSessionBean sessionBean =
(AlphabloxPortletSessionBean)portletRequest.getPortletSession().getAttribute(AlphabloxPortlet.S
ESSION_BEAN);

   // Parameters for edit mode
   PortletData portletData = portletRequest.getData();
   // base name for the blox and name of the bookmark
   String bloxName = (String)portletData.getAttribute("bookmarkName");
   // Flag if page filters should be changed
   boolean pageTarget =
      "true".equals(getParam(portletData.getAttribute("pageTarget")));
   // Flag if row or column sets should be changed
   boolean rowColTarget =
      "true".equals(getParam(portletData.getAttribute("rowColTarget")));
   // Flag if messages should be send as a portlet link
   boolean portletLink =
      "true".equals(getParam(portletData.getAttribute("portletLink")));
   // height of the blox
   String height = (String)portletData.getAttribute("height");
```

```
   // Set dynamic height parameters
   if(height==null || height.equals(""))
      height = "300";
   // Message if there is no bookmark available
   if(bloxName==null) {
      out.write("<b>Please go to the Edit Mode to set up the report.</b>");
      return;
   }
   // Dynamic blox names
   String dataBloxName = bloxName+"_data";
   String presentBloxName = bloxName+"_present";
%>
<%!
// Utility method to get parameters from edit mode
public Object getParam(Object value) {
   if(value==null)
      return "";
   else if(value.equals("null"))
      return "";
   return value;
}
%>
<blox:bookmarks id="bookmarksBlox"/>

<blox:data
    id="dataBlox"
    bloxName="<%=dataBloxName%>" connectOnStartup="false"/>

<blox:present
   id="presentBlox"
   bloxName="<%=presentBloxName%>"
   bookmarkFilter=",name=presentBlox"
   menubarVisible="false"
   visible="false">
   <bloxui:calculationEditor/>
   <blox:data bloxRef="<%=dataBloxName%>"/>
   <bloxportlet:actionLinkDefinition action="<%=AlphabloxPortlet.PORTLET_LINK%>">
    <bloxportlet:parameter name="<%=AlphabloxPortlet.PORTLET_LINK_PARAM%>" />
   </bloxportlet:actionLinkDefinition>
<%
   // Load Bookmark code
   if(bloxName!=null) {

if(bookmarksBlox.bookmarkExists(bloxName,presentBlox.getApplicationName(),"",presentBlox.getBlo
xName(),Bookmark.PUBLIC_VISIBILITY,presentBlox.getBookmarkFilter())) {
         Bookmark bookmark =
bookmarksBlox.getBookmark(bloxName,presentBlox.getApplicationName(),"",presentBlox.getBloxName(
),Bookmark.PUBLIC_VISIBILITY,presentBlox.getBookmarkFilter());
```

```
         BookmarkProperties bookmarkProperties =
bookmark.getBookmarkPropertiesByType(Bookmark.PRESENT_BLOX_TYPE);
         bookmarkProperties.setProperty("dataLayoutAvailable","false");
         presentBlox.loadBookmark(bookmark);
      }
   }

   // Blox to Blox EventHandler
   String targetPresentBlox =
      (String)portletData.getAttribute("targetPresentBlox");
   if(targetPresentBlox!=null)
      presentBlox.getPresentBloxModel().addEventHandler(
         newalphablox.event.BloxToBloxEventHandler(presentBlox,targetPresentBlox));

   // Add PortletLinkEventHandler that handles Click to Action
   if(portletLink)
       presentBlox.getPresentBloxModel().addEventHandler(
       new alphablox.event.PortletLinkEventHandler(presentBlox));

%>
   <% // --> Set Toolbar buttons %>
   <bloxui:toolbar name="<%=ModelConstants.NAVIGATION_TOOLBAR%>" title="" visible="false"/>
   <bloxui:toolbar name="<%=ModelConstants.STANDARD_TOOLBAR%>" title="">
      <bloxui:toolbarButton name="<%=ModelConstants.BOOKMARK_LOAD%>" title=""
visible="false"/>
   </bloxui:toolbar>
   <% // <-- End Set Toolbar buttons %>
</blox:present>
<%
// Receive Portlet Message from Session Bean
String messageText = sessionBean.getValue("message");
if(dataBlox.getMetaData() instanceof MDBMetaData) {
   // Get meta data object
   MDBMetaData meta = (MDBMetaData)dataBlox.getMetaData();
   // Get result set object
   MDBResultSet rs = (MDBResultSet)dataBlox.getResultSet();
   if(messageText!=null) {
      // check if the member name is in MDX format
      if(!messageText.startsWith("["))
         messageText = "["+messageText+"]";
      // resolve member in the meta data object
      Member member = meta.resolveMember(messageText);
      if(member!=null) {
         AxisDimension axisDimension =
rs.resolveAxisDimension(member.getDimension().getDisplayName());
         // Check on what axis the dimension of the member
         // currently is
         if(pageTarget && axisDimension!=null &&
axisDimension.getAxis().getIndex()==Axis.PAGE_AXIS_ID) {
```

```
            // For page only that member should be selected
            dataBlox.setSelectedMembers(new Member[]{member});
         }
         if(rowColTarget && axisDimension!=null &&
axisDimension.getAxis().getIndex()==Axis.ROW_AXIS_ID) {
            // For column or row select the member and its childrem
            Member[] members = member.getChildren();
            Member[] hierarchy = new Member[members.length+1];
            hierarchy[0] = member;
            for(int i=1;i<hierarchy.length;i++)
               hierarchy[i] = members[i-1];
            dataBlox.setSelectedMembers(hierarchy);
         }
      }
   }
}
%>
<DIV style="margin: 6px">
<table width="100%" cellpadding="0" cellspacing="0">
   <tr>
      <td>
         <blox:display
            bloxRef="<%=presentBloxName%>"
            height="<%=height%>"
            width="100%" />
      </td>
   </tr>
</table>
</DIV>
```

Portlet Edit Mode

In this section we show the jsp code for the edit mode of the Alphablox Portlet. The code is depicted in Example A-7.

Example: A-7 AlphabloxPortletedit.jsp

```
<%@ page import="com.alphablox.blox.uimodel.*,
             com.alphablox.blox.uimodel.core.*,
                com.alphablox.blox.repository.*,
                com.alphablox.blox.*,
                org.apache.jetspeed.portlet.*,
                alphablox.*" %>

<%@ taglib uri="/WEB-INF/tlds/blox.tld" prefix="blox" %>
<%@ taglib uri="/WEB-INF/tlds/bloxui.tld" prefix="bloxui"%>
<%@ taglib uri="/WEB-INF/tlds/bloxform.tld" prefix="bloxform"%>
```

```
<%@ taglib uri="/WEB-INF/tlds/bloxlogic.tld" prefix="bloxlogic"%>
<%@ taglib uri="/WEB-INF/tlds/bloxportlet.tld" prefix="bloxportlet"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>

<portletAPI:init/>

<blox:header/>

<DIV>
<%
  // Parameters for edit mode
   PortletData portletData = portletRequest.getData();
   // base name for the blox and name of the bookmark
   String bloxName = (String)portletData.getAttribute("bookmarkName");
   // Flag if page filters should be changed
   String targetPresentBlox =
      (String)getParam(portletData.getAttribute("targetPresentBlox"));
   boolean pageTarget =
      "true".equals(getParam(portletData.getAttribute("pageTarget")));
   // Flag if row or column sets should be changed
   boolean rowColTarget =
      "true".equals(getParam(portletData.getAttribute("rowColTarget")));
   // Flag if messages should be send as a portlet link
   boolean portletLink =
      "true".equals(getParam(portletData.getAttribute("portletLink")));
   // height of the blox
   String height = (String)portletData.getAttribute("height");
   // Set dynamic height parameters
   if(height==null || height.equals(""))
      height = "300";
   // Dynamic blox names
   String dataBloxName = bloxName+"_data";
   String presentBloxName = bloxName+"_present";

  if( portletData!=null ) {
   // --> Bookmark Name entry
   if(bloxName==null) {
%>
  <FORM method="POST" action="<portletAPI:createURI><portletAPI:URIAction
name='<%=AlphabloxPortlet.EDIT_ACTION%>'/></portletAPI:createURI>">
    <LABEL class="wpsLabelText"   for="bookmarkName">Bookmark Name</LABEL><BR>
    <INPUT class="wpsEditField"  name="bookmarkName"
value="<%=getParam(portletData.getAttribute("bookmarkName"))%>" type="text"/><BR>
    <INPUT class="wpsEditField"  name="setBookmarkName" value="true" type="hidden"/><BR>

    <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=AlphabloxPortlet.SUBMIT%>'/>" value="Continue" type="submit"/>
    <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=AlphabloxPortlet.CANCEL%>'/>" value="Cancel" type="submit"/>
```

```
  </FORM>
<%
   } else {
 %>
  <FORM method="POST" action="<portletAPI:createURI><portletAPI:URIAction
name='<%=AlphabloxPortlet.EDIT_ACTION%>'/></portletAPI:createURI>">
    <LABEL class="wpsLabelText"   for="bookmarkName">Bookmark Name</LABEL><BR>
    <INPUT class="wpsEditField"  name="bookmarkName" value="<%=bloxName%>" type="text"/>
    <INPUT class="wpsEditField"  name="setBookmarkName" value="true" type="hidden" />
     
   <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=AlphabloxPortlet.SUBMIT%>'/>" value="Change" type="submit"/>
  </FORM>

  <FORM method="POST" action="<portletAPI:createURI><portletAPI:URIAction
name='<%=AlphabloxPortlet.EDIT_ACTION%>'/></portletAPI:createURI>">
  <INPUT class="wpsEditField"  name="presentBloxName" value="<%=presentBloxName%>"
type="hidden"/>
  <INPUT class="wpsEditField"  name="bookmarkName" value="<%=bloxName%>" type="hidden"/>
  <TABLE width="100%" cellpadding="0" cellspacing="0" border="0">
   <tr>
      <td width="240px">
          <LABEL class="wpsLabelText"   for="portletTitle">Portlet Title</LABEL>
       </td>
       <td>
          <INPUT class="wpsEditField"  name="portletTitle"
value="<%=getParam(portletData.getAttribute("portletTitle"))%>" type="text"/>
       </td>
    </tr>
   <tr>
      <td width="240px">
          <LABEL class="wpsLabelText"   for="height">Height</LABEL>
       </td>
       <td>
       <INPUT class="wpsEditField"  name="height" value="<%=height%>" type="text"/><BR>
    </td>
   </tr>
   <tr>
      <td width="240px">
       <LABEL class="wpsLabelText"   for="targetPresentBlox">Blox to Blox List</LABEL>
       </td>
       <td>
          <INPUT class="wpsEditField"  name="targetPresentBlox" value="<%=targetPresentBlox%>"
type="text"/>
       </td>
    </tr>
    <tr>
    <td width="240px">
      <LABEL class="wpsLabelText"   for="pageTarget">Allow change Page</LABEL>
```

```
    </td>
    <td>
          <INPUT class="wpsEditField"  name="pageTarget" value="true"
<%=(pageTarget)?"checked":""%> type="checkbox"/>
       </td>
   </tr>
   <tr>
      <td width="240px">
          <LABEL class="wpsLabelText"   for="rowColTarget">Allow change Row or Column</LABEL>
      </td>
      <td>
          <INPUT class="wpsEditField"  name="rowColTarget" value="true"
<%=(rowColTarget)?"checked":""%> type="checkbox"/>
       </td>
    </tr>
    <tr>
    <td width="240px">
          <LABEL class="wpsLabelText"   for="portletLink">Enable Portlet Links</LABEL>
      </td>
      <td>
          <INPUT class="wpsEditField"  name="portletLink" value="true"
<%=(portletLink)?"checked":""%> type="checkbox"/>
       </td>
    </tr>
    <tr>
    <td colspan="2">
          <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=AlphabloxPortlet.SUBMIT%>'/>" value="Save" type="submit"/>
      <INPUT class="wpsButtonText" name="<portletAPI:encodeNamespace
value='<%=AlphabloxPortlet.CANCEL%>'/>" value="Cancel" type="submit"/>
      </td>
   </tr>
   </TABLE>
  </FORM>

<blox:bookmarks id="bookmarksBlox"/>

<blox:data
    id="dataBlox"
    bloxName="<%=dataBloxName%>"
    connectOnStartup="false"/>

<blox:present
   id="presentBlox"
   bloxName="<%=presentBloxName%>"
   bookmarkFilter=",name=presentBlox"
   visible="false">
   <bloxui:calculationEditor/>
    <blox:data bloxRef="<%=dataBloxName%>"/>
```

```
</blox:present>
<%
    // Load Bookmark code
   System.out.println(bloxName);
    if(bloxName!=null) {

if(bookmarksBlox.bookmarkExists(bloxName,presentBlox.getApplicationName(),"",presentBlox.getBlo
xName(),Bookmark.PUBLIC_VISIBILITY,presentBlox.getBookmarkFilter())) {
         Bookmark bookmark =
bookmarksBlox.getBookmark(bloxName,presentBlox.getApplicationName(),"",presentBlox.getBloxName(
),Bookmark.PUBLIC_VISIBILITY,presentBlox.getBookmarkFilter());
         BookmarkProperties bookmarkProperties =
bookmark.getBookmarkPropertiesByType(Bookmark.PRESENT_BLOX_TYPE);
         bookmarkProperties.setProperty("dataLayoutAvailable","false");
         presentBlox.loadBookmark(bookmark);
      }
   }
%>

<%
   // Get the PresentBloxModel
   PresentBloxModel model = presentBlox.getPresentBloxModel();
   // Get the standard toolbar
   Toolbar standardToolbar = model.getStandardToolbar();
   // get thenavigation toolbar
   Toolbar navigationToolbar = model.getNavigateToolbar();
   // Show Standard Toolbar
   if(standardToolbar!=null)
      standardToolbar.setVisible(true);
   // Show Navigation Toolbar
   if(navigationToolbar!=null)
      navigationToolbar.setVisible(true);
   model.changed();
   // Show data layout
   presentBlox.setDataLayoutAvailable(true);

   Toolbar toolbar = model.getNavigateToolbar();
   // Add DataSource Selector and Query Button
   if(toolbar.searchForComponent("dataSource")==null) {
      alphablox.component.DataSourceDropDownToolbarButton dataSourceButton = new
alphablox.component.DataSourceDropDownToolbarButton("dataSource");
      dataSourceButton.setDebug(true);
      toolbar.add(dataSourceButton);
      dataSourceButton.setDataBlox(presentBlox.getDataBlox());
      model.addEventHandler(new
alphablox.component.controller.ToolbarController(presentBlox,application.getRealPath("/resource
s/")));
   }
 %>
```

```
   <table width="100%" cellpadding="0" cellspacing="0">
      <tr>
         <td>
            <blox:display bloxRef="<%=presentBloxName%>" height="<%=height%>" width="100%"/>
         </td>
      </tr>
   </table>
<%
   }
  }
else {
  %>Error: PortletData is null.<%
  }
%>
</DIV>
<%!
public Object getParam(Object value) {
   if(value==null)
      return "";
   else if(value.equals("null"))
      return "";
   return value;
}
%>
```

Portlet Controller

In this section, we show the java code for the portlet controller of the Alphablox Portlet. The code is depicted in Example A-8.

Example: A-8 AlphabloxPortlet.java

```
package alphablox;

import java.io.IOException;
import java.io.PrintWriter;
import java.io.Writer;
import java.util.Enumeration;

import org.apache.jetspeed.portlet.*;
import org.apache.jetspeed.portlet.event.*;

import com.alphablox.blox.BloxContext;
import com.alphablox.blox.PresentBlox;
import com.alphablox.blox.RepositoryBlox;
import com.alphablox.blox.ServerBloxException;
import com.alphablox.blox.uimodel.PresentBloxModel;
```

```
import com.alphablox.blox.uimodel.core.ModelException;
import com.alphablox.blox.uimodel.core.Toolbar;

/**
 *
 * A sample portlet based on PortletAdapter
 *
 */
public class AlphabloxPortlet extends PortletAdapter implements ActionListener, MessageListener
,PortletTitleListener {

    public static final String VIEW_JSP         = "/alphablox/jsp/AlphabloxPortletView.";
// JSP file name to be rendered on the view mode
    public static final String EDIT_JSP         = "/alphablox/jsp/AlphabloxPortletEdit.";
// JSP file name to be rendered on the view mode
    public static final String SESSION_BEAN     = "alphablox.AlphabloxPortletSessionBean";
// Bean name for the portlet session
    public static final String FORM_ACTION      = "alphablox.AlphabloxPortletFormAction";
// Action name for the orderId entry form

    public static final String EDIT_ACTION      =
"alphabloxportlet.AlphabloxPortletPortletEditAction";     // Action name for the edit form
    public static final String EDIT_NAME        =
"alphabloxportlet.AlphabloxPortletPortletEditName";       // Attribute name for the PortletData
   public static final String PORTLET_TITLE   = "portletTitle";
// Attribute name for the Portlet Title

    public static final String SUBMIT           = "alphablox.AlphabloxPortletSubmit";
// Parameter name for general submit button
    public static final String CANCEL           = "alphablox.AlphabloxPortletCancel";
// Parameter name for general cancel button

    public static final String PORTLET_LINK = "sendMessage";
   public static final String PORTLET_LINK_PARAM = "messageName";

    /**
    * @see org.apache.jetspeed.portlet.Portlet#init(PortletConfig)
    */
   public void init(PortletConfig portletConfig) throws UnavailableException {
      super.init(portletConfig);
   }

   /**
    * @see org.apache.jetspeed.portlet.PortletAdapter#doView(PortletRequest, PortletResponse)
    */
   public void doView(PortletRequest request, PortletResponse response) throws
PortletException, IOException {
      // Check if portlet session exists
      try {
```

```
      AlphabloxPortletSessionBean sessionBean = getSessionBean(request);
      if( sessionBean==null ) {
          response.getWriter().println("<b>NO PORTLET SESSION YET</b>");
         return;
      }

        // Invoke the JSP to render
       getPortletConfig().getContext().include(VIEW_JSP+getJspExtension(request), request, 
response);
      } catch (Throwable exc) {
           Writer writer = response.getWriter();
           writer.write("<pre>");
           exc.printStackTrace(new PrintWriter(writer));
           writer.write("</pre>");
        }
   }

   public void doEdit(PortletRequest request, PortletResponse response) throws 
PortletException, IOException {
      try {
      // Invoke the JSP to render
       getPortletConfig().getContext().include(EDIT_JSP+getJspExtension(request), request, 
response);
      } catch (Throwable exc) {
           Writer writer = response.getWriter();
           writer.write("<pre>");
           exc.printStackTrace(new PrintWriter(writer));
           writer.write("</pre>");
        }

   }

   /**
    * @see org.apache.jetspeed.portlet.event.ActionListener#actionPerformed(ActionEvent)
    */
   public void actionPerformed(ActionEvent event) throws PortletException {
      if( getPortletLog().isDebugEnabled() )
         getPortletLog().debug("ActionListener - actionPerformed called");
      // ActionEvent handler
      String actionString = event.getActionString();

      PortletRequest request = event.getRequest();
      // Add action string handler here
      AlphabloxPortletSessionBean sessionBean = getSessionBean(request);

      if( EDIT_ACTION.equals(actionString) ) {
         if( request.getParameter(SUBMIT)!=null ) {
             PortletData data = request.getData();
             if( data!=null ) {
```

```
            // --> Loop through parameters
            Enumeration enum = request.getParameterNames();
            while(enum.hasMoreElements()) {
               String key = (String)enum.nextElement();
               String value = request.getParameter(key);
               if(key!=null && value!=null && !value.equals("null"))
                  data.setAttribute(key,value);
            }
            // <-- End Look through Parameters

            try {
              data.store();// save to data store

              // --> Save Bookmark code
              String presentBloxName = request.getParameter("presentBloxName");
              String bookmarkName = request.getParameter("bookmarkName");

              if(presentBloxName!=null) {
               BloxContext bloxContext =
(BloxContext)request.getSession().getAttribute(BloxContext.BLOX_CONTEXT_ATTR);
               if(bloxContext!=null) {
                  PresentBlox presentBlox =
(PresentBlox)bloxContext.getBlox(presentBloxName);
                  try {

presentBlox.saveBookmarkHidden(RepositoryBlox.VISIBILITY_APPLICATION,"",bookmarkName,"");
                     presentBlox.setDataLayoutAvailable(false);
                     PresentBloxModel model = presentBlox.getPresentBloxModel();
                     Toolbar standardToolbar = model.getStandardToolbar();
                     Toolbar navigationToolbar = model.getNavigateToolbar();
                     if(standardToolbar!=null)
                        standardToolbar.setVisible(false);
                     if(navigationToolbar!=null)
                        navigationToolbar.setVisible(false);
                     model.changed();
                     bloxContext.deleteBlox(presentBlox);
                     bloxContext.deleteBlox(presentBlox.getDataBlox());

                  } catch (ServerBloxException e) {
                     getPortletLog().error(e.getMessage());
                  } catch (ModelException e) {
                     getPortletLog().error(e.getMessage());
                  }
               }
              }
              // If the bookmark doesn't get set forward the request to the view mode again
              if(request.getParameter("setBookmarkName")==null)
               event.getRequest().setModeModifier(ModeModifier.PREVIOUS );
```

```
                // <-- End Save Bookmark

              }
              catch (IOException ioe) {
                 if( getPortletLog().isErrorEnabled() )
                    getPortletLog().error("Error on PortletData.store():
"+ioe.getMessage());
              }
           }
        }
     }
     if (actionString.equals(PORTLET_LINK)) {
          String portletLinkMember = request.getParameter(PORTLET_LINK_PARAM);
       PortletMessage message = new DefaultPortletMessage(portletLinkMember);
          getPortletConfig().getContext().send(null,message);
      }
  }

  /**
   * @see org.apache.jetspeed.portlet.event.MessageListener#messageReceived(MessageEvent)
   */
  public void messageReceived(MessageEvent event) throws PortletException {
      if( getPortletLog().isDebugEnabled() )
          getPortletLog().debug("MessageListener - messageReceived called");
      // MessageEvent handler
      PortletMessage msg = event.getMessage();
      // Add PortletMessage handler here
      if( msg instanceof DefaultPortletMessage ) {
          String messageText = ((DefaultPortletMessage)msg).getMessage();
          AlphabloxPortletSessionBean sessionBean =
           getSessionBean(event.getRequest());
          sessionBean.setValue("message",messageText);
      }
      else {
          // Add general PortletMessage handler here
      }
  }

  /**
   * Get SessionBean.
   *
   * @param request PortletRequest
   * @return AlphabloxPortletSessionBean
   */
  private AlphabloxPortletSessionBean getSessionBean(PortletRequest request) {
       PortletSession session = request.getPortletSession(false);
       if( session == null )
       return null;
```

```
        AlphabloxPortletSessionBean sessionBean =
(AlphabloxPortletSessionBean)session.getAttribute(SESSION_BEAN);
        if( sessionBean == null ) {
        sessionBean = new AlphabloxPortletSessionBean();
        session.setAttribute(SESSION_BEAN,sessionBean);
        }
        return sessionBean;
   }

   /**
    * Returns the file extension for the JSP file
    *
    * @param request PortletRequest
    * @return JSP extension
    */
   private static String getJspExtension(PortletRequest request) {
      String markupName = request.getClient().getMarkupName();
      return "jsp";
   }

   public void doTitle(PortletRequest request, PortletResponse response) throws
PortletException, IOException {
      PortletSettings pSettings = request.getPortletSettings();
      String title = (String)request.getData().getAttribute(PORTLET_TITLE);
      if(title == null || title.equals("") || title.equals("null"))
      {
         java.util.Locale locale = request.getLocale();
         Client client = request.getClient();
         title = pSettings.getTitle(locale, client);
      }
      response.getWriter().print(title);
   }
}
```

Portlet Session Bean

In this section, we show the java code for the session bean of the Alphablox Portlet. This portlet stores name value pairs in the session object. The code is depicted in Example A-9.

Example: A-9 AlphabloxPortletSessionBean.java

```
package alphablox;

import java.util.HashMap;

/**
 *
```

```
 * A sample Java bean that stores portlet instance data in portlet session.
 *
 */
public class AlphabloxPortletSessionBean {

   //**********************************************************
   //* Last text for the text form
   //**********************************************************

    private String formText = "";
    private HashMap hashMap = new HashMap();

    public void setValue(String param,String value) {
    this.hashMap.put(param,value);
    }

    public String getValue(String param) {
    return (String)this.hashMap.get(param);
    }
   /**
    * Set last text for the text form.
    *
    * @param formText last text for the text form.
    */
    public void setFormText(String formText) {
        this.formText = formText;
    }

   /**
    * Get last text for the text form.
    *
    * @return last text for the text form
    */
    public String getFormText() {
        return this.formText;
    }

}
```

Blox-To-Blox EventHandler

In this section, we show the java code for the Blox-to-Blox Event Handler, It collects the metadata of the source blox on a double-click and controlls the filters of the target blox. The code is depicted in Example A-10 on page 409.

Example: A-10 BloxToBloxEventHandler.java

```
/*
 * (c) Copyright IBM Corp. 2003, 2004
 * Author Robert Frankus
 * Created on May 19, 2005
 */
package alphablox.event;

import java.util.*;

import com.alphablox.blox.*;
import com.alphablox.blox.data.mdb.*;
import com.alphablox.blox.data.mdb.Axis;
import com.alphablox.blox.uimodel.*;
import com.alphablox.blox.uimodel.core.*;
import com.alphablox.blox.uimodel.core.event.*;
import com.alphablox.blox.uimodel.core.grid.GridCell;

public class BloxToBloxEventHandler implements IEventHandler {
   PresentBlox presentBlox;
   // name of the target presentBlox(s)
   String targetPresentBlox;
   // Put all the coordinates into a Hashmap
    Hashtable memberHash = new Hashtable();


   public BloxToBloxEventHandler(PresentBlox presentBlox,
         String targetPresentBlox) throws Exception {
      this.presentBlox = presentBlox;
      this.targetPresentBlox = targetPresentBlox;
   }

   public boolean handleDoubleClickEvent(DoubleClickEvent event)
      throws Exception {
       Component component = event.getComponent();
       // Check if double click happened on a GridCell
       if(component instanceof GridBrixCellModel) {
          GridCell gridCell = (GridCell) component;
          // Check if the cell is a data cell
          if(!gridCell.isRowHeader()&& !gridCell.isColumnHeader()) {
             getMembers();
            try {
                StringTokenizer stringTokenizer =
                     new StringTokenizer(this.targetPresentBlox,",");
                // Get the BloxContext to find the other blox
                BloxContext bloxContext = presentBlox.getBloxContext();
                while(stringTokenizer.hasMoreTokens()) {
                String token = stringTokenizer.nextToken();
```

```
                    String bloxName = token+"_present";
                    // Check for the targetPresentBlox if member exists
                    // and on what axis
                    PresentBlox blox =
                      (PresentBlox)bloxContext.getBlox(bloxName);
                    MDBResultSet rs =
                      (MDBResultSet) blox.getDataBlox().getResultSet();
                    MDBMetaData meta =
                      (MDBMetaData) blox.getDataBlox().getMetaData();
                    if(blox!=null) {
                      Enumeration dimensionNames = memberHash.keys();
                      while(dimensionNames.hasMoreElements()) {
                          String dimensionName =
                             (String)dimensionNames.nextElement();
                          String memberName =
                             (String) memberHash.get(dimensionName);
                          Member member = meta.resolveMember(memberName);
                          AxisDimension axisDimension =
                             rs.resolveAxisDimension(dimensionName);
                          // Consider only members that are on page
                          if(axisDimension!=null &&
                          axisDimension.getAxis().getIndex()==Axis.PAGE_AXIS_ID) {
                             if(member!=null)
                                blox.getDataBlox().setSelectedMembers(
                                   new Member[]{member});
                          }
                      }
                    }
                  }
              } catch(Exception e) {
                    MessageBox.message( component, "Error",
                      "Error:"+e.getMessage());
              }
              return true;
            }
        }
        return false;
    }

    public String stringFrom(String[] array){
         String buf = "";
         for (int i = 0; i < array.length; i++) {
             buf += (i == 0 ? "" : ", ") + array[i];
         }
         return buf;
     }

    public Hashtable getMembers() throws ServerBloxException {
       // Get GridBrixModel
```

```
GridBrixModel grid = presentBlox.getPresentBloxModel().getGrid();
// get MDB Result Set
MDBResultSet rs = (MDBResultSet)
   presentBlox.getDataBlox().getResultSet();
// Get all selected cells
GridCell[] gridCells = grid.getSelectedCells();

// loop through all selected cells
for (int i = 0; i < gridCells.length; i++) {
   GridCell gridCell = gridCells[i];
   // get the corresponding object to the GridCell in the MDBResltSet
   Object object = grid.findGridBrixCell(rs,gridCell);

   // If the Object is a data cell then cast to Cell
   if(object instanceof Cell) {
      Cell cell = (Cell) object;
      // Get the row and column tuples for the cell
      Tuple[] tuples = cell.getTuples();
      for (int j = 0; j < tuples.length; j++) {
         Tuple tuple = tuples[j];
         TupleMember[] members = tuple.getMembers();
         for (int k = 0; k < members.length; k++) {
            TupleMember member = members[k];
            // exclude calculated members
            if(!member.isCalculatedMember()) {
               String uniqueMember = member.getUniqueName();
               // in case the member is a shared Essbase member
               // use the display name
               if(uniqueMember.indexOf("\u0001")>-1)
                  uniqueMember = member.getDisplayName();
               // Add member to hash map
               memberHash.put(member.getDimension().getDisplayName(),
                        uniqueMember);
            }
         }
      }
      // Also add all page members
      Axis pageAxis = rs.getAxis(Axis.PAGE_AXIS_ID);
      for(int j=0;j<pageAxis.getTupleCount();j++) {
         Tuple tuple = pageAxis.getTuple(j);
         for(int k=0;k<tuple.getMemberCount();k++) {
            TupleMember member = tuple.getMember(k);
            memberHash.put(
               member.getDimension().getDisplayName(),
               member.getUniqueName().substring(
                  member.getUniqueName().indexOf(".")+1));
         }
      }
      // Last get the cube name and add it to the hash map
```

```
                Cube cube = rs.getCubes()[0];
                memberHash.put("cube",cube.getName());
            }
            // if the object is a Tuple Member
            else if(object instanceof TupleMember) {
                TupleMember tupleMember = (TupleMember)object;
                if(gridCell.isRowHeader()) {
                    memberHash.put(
                        tupleMember.getDimension().getDisplayName(),
                        tupleMember.getUniqueName());
                    Axis pageAxis = tupleMember.getTuple().getAxis().
                        getResultSet().getAxis(Axis.PAGE_AXIS_ID);
                    for(int j=0;j<pageAxis.getTupleCount();j++) {
                        Tuple tuple = pageAxis.getTuple(j);
                        for(int k=0;k<tuple.getMemberCount();k++) {
                            TupleMember member = tuple.getMember(k);
                            memberHash.put(
                                member.getDimension().getDisplayName(),
                                member.getUniqueName().substring(
                                    member.getUniqueName().indexOf(".")+1));
                        }
                    }
                }
            }
        }
        return memberHash;
    }
}
```

Portlet Link Event Handler

In this section we show the java code for the portlet link event handler. It sends the shipper information from the chart to the other portlets using portlet messages. The code is depicted in Example A-11.

Example: A-11 PortletLinkEventHandler.java

```
/*
 * (c) Copyright IBM Corp. 2006
 * Author Robert Frankus
 * Created on May 19, 2005
 */
package alphablox.event;

import alphablox.AlphabloxPortlet;

import com.alphablox.blox.PresentBlox;
```

```
import com.alphablox.blox.data.MetaData;
import com.alphablox.blox.data.mdb.MDBMetaData;
import com.alphablox.blox.data.mdb.Member;
import com.alphablox.blox.portlet.PortletLink;
import com.alphablox.blox.uimodel.ChartBrixModel;
import com.alphablox.blox.uimodel.core.*;
import com.alphablox.blox.uimodel.core.chart.*;
import com.alphablox.blox.uimodel.core.chart.common.*;
import com.alphablox.blox.uimodel.core.event.DoubleClickEvent;
import com.alphablox.blox.uimodel.core.event.IEventHandler;

public class PortletLinkEventHandler implements IEventHandler {
   // Source PresentBlox
   PresentBlox presentBlox;
   // Portlet Link object specified as nested blox of the PresentBlox
   PortletLink portletLink;

   public PortletLinkEventHandler(PresentBlox presentBlox) {
      this.presentBlox = presentBlox;
      this.portletLink =
         presentBlox.getPortletLink(AlphabloxPortlet.PORTLET_LINK);
   }

   public boolean handleDoubleClickEvent(DoubleClickEvent event)
      throws Exception {
      Component component = event.getComponent();
      // Check if clicked component is a Chart
      if(component instanceof Chart) {
         Chart theChart = (Chart) event.getComponent();
         // get the selected component in the chart
         ChartComponent chartComponent = theChart.getSelectedChartComponent();
         // Check if the selected component is a single data
         // series, a bar
         if (chartComponent instanceof SingleValueDataSeries) {
            ChartBrixModel cbModel =
               presentBlox.getPresentBloxModel().getChart();
            String memberName = null;
            SingleValueDataSeries series =
               (SingleValueDataSeries) chartComponent;
            int selectedIndex = series.getSelectedIndex();
            /* Only Possible in V8.4 */
            // Get the Navtive Data Point this is only available in V8.4
            ChartDataPoint nativeDataPoint =
               series.getNativeDataPoint(selectedIndex);
            // Get series members, i.e. legend
            String[] seriesMembers =
               cbModel.getUniqueSeriesMembers(nativeDataPoint);
            // Get group members, i.e. x-axis
            String[] groupMembers =
```

```
                    cbModel.getUniqueGroupMembers(nativeDataPoint);
                 // Get filter members
                 String[] filterMembers = cbModel.getUniqueFilterMembers();
                 // Get the first member on the x-axis, shipper
                 // in this case we know there is only one dimension on
                 // the x-axis
                 memberName = groupMembers[0];
                 // resolve the member in the MetaData object to get the
                 // display name
                 MetaData metaData = presentBlox.getDataBlox().getMetaData();
                 if(metaData instanceof MDBMetaData) {
                    MDBMetaData mdbMetaData = (MDBMetaData)metaData;
                    Member member = mdbMetaData.resolveMember(memberName);
                    memberName = member.getDisplayName();
                 }
                 /* Only Possible in V8.4 End*/
                 /*
                 * Prior version 8.4 it is necessary to get the member name
                 * from the labels
                 Label[] labels = series.getOrdinalAxis().getLabels();
                 memberName = labels[selectedIndex].getDisplayText();
                 */
                 portletLink.setParameterValue(
                    AlphabloxPortlet.PORTLET_LINK_PARAM,
                    "ChangeShipper:"+memberName);
                 cbModel.getDispatcher().sendClientCommand(
                    portletLink.getLinkCall());
                    return true;
                }
             }
              return false;
          }
}
```

Glossary

Access Control List (ACL). The list of principals that have explicit permission (to publish, to subscribe to, and to request persistent delivery of a publication message) against a topic in the topic tree. The ACL defines the implementation of topic-based security.

Aggregate. Pre-calculated and pre-stored summaries, kept in the data warehouse to improve query performance. A multidimensional summary table derived from InfoCube data for performance; can be stored in RDBMS or MS Analysis Services.

Aggregation. An attribute level transformation that reduces the level of detail of available data. For example, having a Total Quantity by Category of Items rather than the individual quantity of each item in the category.

Alert. A message that indicates a processing situation that requires specific and immediate attention.

AMI. See Application Messaging Interface.

Application Link Enabling. Supports the creation and operation of distributed applications. And, application integration is achieved via synchronous and asynchronous communication, not via a central database. Provides business-controlled message exchange with consistent data on loosely linked SAP applications.

Application Messaging Interface. The programming interface provided by MQSeries that defines a high level interface to message queuing services.

Application Programming Interface. An interface provided by a software product that enables programs to request services.

Asynchronous Messaging. A method of communication between programs in which a program places a message on a message queue, then proceeds with its own processing without waiting for a reply to its message.

Attribute. A field in a dimension table.

Basis. A set of middleware programs and tools from SAP that provides the underlying base that enables applications to be seamlessly interoperable and portable across operating systems and database products.

BEx. Business Explorer: the SAP query and reporting tool for users tightly coupled with BW.

BLOB. Binary Large Object, a block of bytes of data (for example, the body of a message) that has no discernible meaning, but is treated as one solid entity that cannot be interpreted.

BLOX. DB2 Alphablox software components.

Broker domain. A collection of brokers that share a common configuration, together with the single Configuration Manager that controls them.

Characteristic. A business intelligence dimension.

Central Processing Unit. Also called a CPU. It is the device (a chip or circuit board for example) that houses processing capability of a computer. That processing capability is the execution of instructions to perform specific functions. There can be multiple CPUs in a computer.

Cluster. A group of records with similar characteristics. In WebSphere MQ, a group or two or more queue managers on one or more computers, providing programmatic interconnection, and allowing queues to be shared amongst them for load balancing and redundancy.

Compensation. The ability of DB2 to process SQL that is not supported by a data source on the data from that data source.

Commit. An operation that applies all the changes made during the current unit of recovery or unit of work. After the operation is complete, a new unit of recovery or unit of work begins.

Composite Key. A key in a fact table that is the concatenation of the foreign keys in the dimension tables.

Computer. A programmable device that consists of memory, CPU, and storage capability. It has an attached device to input data and instructions, and a device to output results.

Configuration Manager. A component of WebSphere MQ Integrator that acts as the interface between the configuration repository and an existing set of brokers.

Configuration repository. Persistent storage for broker configuration and topology definition.

Configuration. The collection of brokers, their execution groups, the message flows and sets that are assigned to them, and the topics and associated access control specifications.

Connector. See Message processing node connector.

Control Center. The graphical interface that provides facilities for defining, configuring, deploying, and monitoring resources of the WebSphere MQ Integrator network.

Dashboard. A business information interface that displays business intelligence with easy-to-comprehend graphical icons.

Data Append. A data loading technique where new data is added to the database leaving the existing data unaltered.

Data Cleansing. A process of data manipulation and transformation to eliminate variations and inconsistencies in data content. This is to improve the quality, consistency, and usability of the data.

Data Federation. The process of enabling data from multiple heterogeneous data sources to appear as if it is contained in a single relational database. Can also be referred to “distributed access”.

Data Mart. An implementation of a data warehouse, with a smaller and more tightly restricted scope, such as for a department or workgroup. It could be independent or derived from another data warehouse environment.

Data Mining. A mode of data analysis that has a focus on the discovery of new information, such as unknown facts, data relationships, or data patterns.

Data Partition. A segment of a database that can be accessed and operated on independently even though it is part of a larger data structure.

Data Refresh. A data loading technique where all the data in a database is completely replaced with a new set of data.

Data Warehouse. A specialized data environment developed, structured, and used specifically for decision support and informational applications. It is subject-oriented rather than application-oriented. Data is integrated, non-volatile, and time variant.

Database Instance. A specific independent implementation of a DBMS in a specific environment. For example, there might be an independent DB2 DBMS implementation on a Linux server in Boston supporting the Eastern offices, and another separate and independent DB2 DBMS on the same Linux server supporting the Western offices. They would represent two instances of DB2.

Database Partition. Part of a database that consists of its own data, indexes, configuration files, and transaction logs.

DB Connect. Enables connection to several relational database systems and the transfer of data from these database systems into the SAP Business Information Warehouse.

Debugger. A facility on the Message Flows view in the Control Center that enables message flows to be visually debugged.

Deploy. Make operational the configuration and topology of the broker domain.

Dimension. Data that further qualifies and/or describes a measure, such as amounts or durations.

Distributed Application In message queuing, a set of application programs that can each be connected to a different queue manager, but that collectively constitute a single application.

Distribution list A list of MQSeries queues to which a message can be put using a single statement.

Drill-down. Iterative analysis, exploring facts at more detailed levels of the dimension hierarchies.

Dynamic SQL. SQL that is interpreted during execution of the statement.

Element. A unit of data within a message that has a business meaning.

Enqueue. To put a message on a queue.

Enterprise Application Integration. A Message-based, transaction-oriented, point-to-point (or point-to-hub) brokering and transformation for application-to-application integration.

Enterprise Information Integration. Optimized and transparent data access and transformation layer providing a single relational interface across all enterprise data.

Enrichment. The creation of derived data. An attribute level transformation performed by some type of algorithm to create one or more new (derived) attributes.

ETL - Extract Transform Load. Set-oriented, point-in-time transformation for migration, consolidation, and data warehousing.

Event. A signal to the background processing system that a certain status has been reached in the SAP system. The background processing system then starts all processes that were waiting for this event.

Event Queue. The queue onto which the queue manager puts an event message after it detects an event. Each category of event (queue manager, performance, or channel event) has its own event queue.

Execution group. A named grouping of message flows that have been assigned to a broker.

Extenders. These are program modules that provide extended capabilities for DB2 and are tightly integrated with DB2.

FACTS. A collection of measures, and the information to interpret those measures in a given context.

Federated Server. Any DB2 server where the DB2 Information Integrator is installed.

Federation. Providing a unified interface to diverse data.

Foreign Key. The combination of one or more columns within a table that reference (are identical to) the primary key column(s) of another table.

Framework. In WebSphere MQ, a collection of programming interfaces that allows customers or vendors to write programs that extend or replace certain functions provided in WebSphere MQ products.

Gateway. A means to access a heterogeneous data source. It can use native access or ODBC technology.

Grain. The fundamental lowest level of data represented in a dimensional fact table.

Hash Partitioning. Data for a table is distributed across the partitions of a database, based on a hashing algorithm that is applied to a set of columns within the table.

Input node. A message flow node that represents a source of messages for the message flow.

Instance. A complete database environment.

IVews. InfoObjects (master data) with properties, text, and hierarchies.

Java Database Connectivity. An application programming interface that has the same characteristics as ODBC but is specifically designed for use by Java database applications.

Java Development Kit. Software package used to write, compile, debug, and run Java applets and applications.

Java Message Service. An application programming interface that provides Java language functions for handling messages.

Java Runtime Environment. A subset of the Java Development Kit that allows you to run Java applets and applications.

Key Performance Indicator (KPI). A specific value threshold of a business metric that defines the acceptable business performance level.

Listener. In WebSphere MQ distributed queuing, a program that detects incoming network requests and starts the associated channel.

Master data. The facts describing your core business entities: customers, suppliers, partners, products, materials, bill of materials, chart of accounts, location, and employees.

Master data integration (MDI). The information integration capabilities necessary to solve the broadest range of MDM implementation and ongoing operational challenges across any industry, business function, and scope of business data.

Master data management (MDM). The set of disciplines, technologies, and solutions used to create and maintain consistent, complete, contextual, and accurate business data for all stakeholders (such as users, applications, data warehouses, processes, companies, trading partners, and customers) across and beyond the enterprise.

Materialized Query Table. A table where the results of a query are stored, for later reuse.

Measure. A data item that measures the performance or behavior of business processes.

Message broker. A set of execution processes hosting one or more message flows.

Message domain. The value that determines how the message is interpreted (parsed).

Message flow. A directed graph that represents the set of activities performed on a message or event as it passes through a broker. A message flow consists of a set of message processing nodes and message processing connectors.

Message parser. A program that interprets the bit stream of an incoming message and creates an internal representation of the message in a tree structure. A parser is also responsible to generate a bit stream for an outgoing message from the internal representation.

Message processing node connector. An entity that connects the output terminal of one message processing node to the input terminal of another.

Message processing node. A node in the message flow, representing a well defined processing stage. A message processing node can be one of several primitive types or it can represent a subflow.

Message Queue Interface. The programming interface provided by the WebSphere MQ queue managers. This programming interface allows application programs to access message queuing services.

Message queuing. A communication technique that uses asynchronous messages for communication between software components.

Message repository. A database that holds message template definitions.

Message set. A grouping of related messages.

Message type. The logical structure of the data within a message.

Metadata. Commonly called data (or information) about data. It describes or defines data elements.

Metrics (business). Measurements of business performance.

MOLAP. Multi-dimensional OLAP. Can be called MD-OLAP. It is OLAP that uses a multi-dimensional database as the underlying data structure.

MultiCube. A pre-joined view of two or more cubes represented as an OLAP cube to the user.

MQSeries. A previous name for WebSphere MQ.

Multidimensional analysis. Analysis of data along several dimensions. For example, analyzing revenue by product, store, and date.

Nickname. An identifier that is used to reference the object located at the data source that you want to access.

Node. A device connected to a network.

Node Group. Group of one or more database partitions.

ODS. Operational data store: A relational table for holding clean data to load into InfoCubes and can support some query activity.

OLAP. OnLine Analytical Processing. Multidimensional data analysis, performed in real time. Not dependent on underlying data schema.

Open Database Connectivity. A standard application programming interface for accessing data in both relational and non-relational database management systems. Using this API, database applications can access data stored in database management systems on a variety of computers even if each database management system uses a different data storage format and programming interface. ODBC is based on the call level interface (CLI) specification of the X/Open SQL Access Group.

Open Hub. Enables distribution of data from an SAP BW system for external uses.

Optimization. The capability to enable a process to execute and perform in such a way as to maximize performance, minimize resource utilization, and minimize the process execution response time delivered to the user.

Output node. A message processing node that represents a point at which messages flow out of the message flow.

Partition. Part of a database that consists of its own data, indexes, configuration files, and transaction logs.

Pass-through. The act of passing the SQL for an operation directly to the data source without being changed by the federation server.

Pivoting. Analysis operation where user takes a different viewpoint of the results. For example, by changing the way the dimensions are arranged.

Plug-in node. An extension to the broker, written by a third-party developer, to provide a new message processing node or message parser in addition to those supplied with the product.

Point-to-point. Style of application messaging in which the sending application knows the destination of the message.

Predefined message. A message with a structure that is defined before the message is created or referenced.

Process. An activity within or outside an SAP system with a defined start and end time.

Process Variant. Name of the process. A process can have different variants. For example, in the loading process, the name of the InfoPackage represents the process variants. The user defines a process variant for the scheduling time.

Primary Key. Field in a database table that is uniquely different for each record in the table.

PSA. Persistent staging area: flat files that hold extract data that has not yet been cleaned or transformed.

Pushdown. The act of optimizing a data operation by pushing the SQL down to the lowest point in the federated architecture where that operation can be executed. More simply, a pushdown operation is one that is executed at a remote server.

Queue Manager. A subsystem that provides queuing services to applications. It provides an application programming interface so that applications can access messages on the queues that are owned and managed by the queue manager.

Queue. A WebSphere MQ object. Applications can put messages on, and get messages from, a queue. A queue is owned and managed by a queue manager. A local queue is a type of queue that can contain a list of messages waiting to be processed. Other types of queues cannot contain messages but are used to point to other queues.

Range Partitioning. Data for a table is distributed across the partitions of a database, where each partition only contains a specific “range” of data.

Referential Integrity. Database capability that ensures that relationships between tables remain consistent.

RemoteCube. An InfoCube whose transaction data is managed externally rather than in SAP BW.

ROLAP. Relational OLAP. Multidimensional analysis using a multidimensional view of relational data. A relational database is used as the underlying data structure.

Roll-up. Iterative analysis, exploring facts at a higher level of summarization.

Schema. Collection of database objects and components under the control of (owned by) a specific userid or name.

Server. A device or computer that manages network resources, such as printers, files, databases, and network traffic.

Shared disk. All database instances access a common database. The SQL statement is shipped to a single instance and the data is passed between the nodes as necessary to process the individual SQL statements that each database instance is processing.

Shared nothing. A data management architecture where nothing is shared between processes. Each process has its own processor, memory, and disk space.

Slice and Dice. Analysis across several dimensions and across many categories of data items to uncover business behavior and rules.

SOAP. Defines a generic message format in XML.

Star Schema. A data warehouse schema, consisting of a single “fact table" with a compound primary key, with one segment for each "dimension" and with additional columns of additive, numeric facts.

Static SQL. SQL that has been compiled prior to execution to provide best performance.

Subflow. A sequence of message processing nodes that can be included within a message flow.

Subject Area. A logical grouping of data by categories, such as customers or items.

Surrogate Key. An artificial or synthetic key that is used as a substitute for a natural key.

Synchronous Messaging. A method of communication between programs in which a program places a message on a message queue and then waits for a reply before resuming its own processing.

Thread. In WebSphere MQ, the lowest level of parallel execution available on an operating system platform.

Type Mapping. The mapping of a specific data source type to a DB2 UDB data type.

UDDI. A special Web service which allows users and applications to locate Web services.

Unit of Work. A recoverable sequence of operations performed by an application between two points of consistency.

User Mapping. An association made between the federated server user ID and password and the data source (to be accessed) used ID and password.

Virtual Database. A federation of multiple heterogeneous relational databases.

Warehouse Catalog. A subsystem that stores and manages all the system metadata.

WebSphere MQ. A family of IBM licensed programs that provides message queuing services.

Workbook. Microsoft Excel workbook with references to InfoProvider.

Wrapper. The means by which a data federation engine interacts with heterogeneous sources of data. Wrappers take the SQL that the federation engine uses and map it to the API of the data source to be accessed. For example, they take DB2 SQL and transform it to the language understood by the data source to be accessed.

WSDL. Language to define specific SOAP message interfaces understood by the Web services provider.

XML. Defines a universal way of representing data, and an XML schema defines the format.

xtree. A query-tree tool that allows you to monitor the query plan execution of individual queries in a graphical environment.

zero latency. This is a term applied to a process where there are no delays as it goes from start to completion.

Abbreviations and acronyms

ACS	access control system
ADK	Archive Development Kit
AIX	Advanced Interactive eXecutive from IBM
ALE	Application Link Enabling
AMI	Application Messaging Interface
API	Application Programming Interface
AQR	automatic query rewrite
AR	access register
ARM	automatic restart manager
ART	access register translation
ASCII	American Standard Code for Information Interchange
AST	Application Summary Table
ATM	Asynchronous Transfer Mode
BAPI	Business Application Programming Interface
BAS	Business Application Services
BI	Business Intelligence
BIRA	Business Integration Reference Architecture
BIW	Business Information Warehouse (SAP)
BLOB	Binary Large OBject
BPEL	Business Process Execution Language
BPM	Business Performance Management
BSM	Business Service Management
BW	Business Information Warehouse (SAP)
CBE	Common Business Event
CCMS	Computing Center Management System
CEI	Common Event Infrastructure
CIO	Chief Information Officer
CLI	Call Level Interface
CLOB	Character Large OBject
CLP	Command Line Processor
CORBA	Common Object Request Broker Architecture
CPM	Corporate Performance Management
CPU	Central Processing Unit
CSA	Common Storage Area
CS-WS	Conversation Support for Web Services
DADx	Document Access Definition Extension
DB	Database
DBA	Database Administrator
DB2	Database 2
DB2II	DB2 Information Integrator
DB2 UDB	DB2 Universal DataBase
DB2 II	DB2 Information Integrator
DBMS	DataBase Management System
DCE	Distributed Computing Environment
DCM	Dynamic Coserver Management
DCOM	Distributed Component Object Model
DDL	Data Definition Language
DLL	Dynamically Linked Library

DIMID	Dimension Identifier
DML	Data Manipulation Language
DNS	Domain Name System
DRDA	Distributed Relational Database Architecture
DSA	Dynamic Scalable Architecture
DSN	Data Source Name
DSS	Decision Support System
EAI	Enterprise Application Integration
EAR	Enterprise Archive
EBCDIC	Extended Binary Coded Decimal Interchange Code
EDA	Enterprise Data Architecture
EDU	Engine Dispatchable Unit
EGM	Enterprise Gateway Manager
EII	Enterprise Information Integration
EIS	Enterprise Information System
EJB	Enterprise Java Beans
EPM	Enterprise Performance Management
ER	Enterprise Replication
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
ESE	Enterprise Server Edition
ETL	Extract Transform and Load
FDL	Flow Definition Language
FP	FixPak
FTP	File Transfer Protocol
Gb	Gigabits
GB	Giga Bytes
GUI	Graphical User Interface
HDR	High availability Data Replication
HPL	High Performance Loader
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
I/O	Input/Output
IBM	International Business Machines Corporation
ID	Identifier
IDE	Integrated Development Environment
IDS	Informix Dynamic Server
II	Information Integration
IIOP	Internet Inter-ORB Protocol
IMG	Integrated Implementation Guide (for SAP)
IMS	Information Management System
I/O	input/output
ISAM	Indexed Sequential Access Method
ISM	Informix Storage Manager
ISV	Independent Software Vendor
IT	Information Technology
ITR	Internal Throughput Rate
ITSO	International Technical Support Organization
IX	Index
J2C	J2EE Connector
J2EE	Java 2 Platform Enterprise Edition
JAR	Java Archive
JDBC	Java DataBase Connectivity
JDK	Java Development Kit
JE	Java Edition
JMS	Java Message Service
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment

JSP	JavaServer Pages
JSR	Java Specification Requests
JTA	Java Transaction API
JVM	Java Virtual Machine
KB	Kilobyte (1024 bytes)
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
LOB	Line of Business
LPAR	Logical Partition
LV	Logical Volume
Mb	Mega bits
MB	Mega Bytes (1,048,576 bytes)
MD	Master Data
MDC	MultiDimensional Clustering
MDI	Master Data Integration
MDM	Master Data Management
MIS	Management Information System
MQAI	WebSphere MQ Administration Interface
MQI	Message Queuing Interface
MQSC	WebSphere MQ Commands
MVC	Model-View-Controller
MQT	Materialized Query Table
MPP	Massively Parallel Processing
MRM	Message Repository Manager
NPI	Non-Partitioning Index
ODA	Object Discovery Agent
ODBC	Open DataBase Connectivity
ODS	Operational Data Store
OLAP	OnLine Analytical Processing
OLE	Object Linking and Embedding
OLTP	OnLine Transaction Processing
ORDBMS	Object Relational DataBase Management System
OS	Operating System
PDS	Partitioned Data Set
PIB	Parallel Index Build
PSA	Persistent Staging Area
RBA	Relative Byte Address
RDBMS	Relational DataBase Management System
RID	Record Identifier
RMI	Remote Method Invocation
RR	Repeatable Read
RS	Read Stability
SAX	Simple API for XML
SDK	Software Developers Kit
SID	Surrogate Identifier
SMIT	Systems Management Interface Tool
SMP	Symmetric MultiProcessing
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SPL	Stored Procedure Language
SQL	Structured Query
TMU	Table Management Utility
TS	Tablespace
UDB	Universal DataBase
UDB	Universal DataBase
UDDI	Universal Description, Discovery and Integration of Web Services
UDF	User Defined Function
UDR	User Defined Routine
UML	Unified Modeling Language
URL	Uniform Resource Locator
VG	Volume Group (Raid disk terminology).

VLDB	Very Large DataBase
VSAM	Virtual Sequential Access Method
VTI	Virtual Table Interface
W3C	World Wide Web Consortium
WAR	Web Archive
WAS	WebSphere Application Server
WBI	WebSphere Business Integration
WfMC	Workflow Management Coalition
WLM	Workload Management
WORF	Web services Object Runtime Framework
WPS	WebSphere Portal Server
WSAD	WebSphere Studio Application Developer
WSDL	Web Services Description Language
WSFL	Web Services Flow Language
WS-I	Web Services Interoperability Organization
WSIC	Web Services Choreography Interface
WSIF	Web Services Invocation Framework
WSIL	Web Services Inspection Language
WSMF	Web Services Management Framework
WWW	World Wide Web
XBSA	X-Open Backup and Restore APIs
XML	eXtensible Markup Language
XSD	XML Schema Definition

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 428. Note that some of the documents referenced here might be available in softcopy only.

- *Business Performance Management . . . Meets Business Intelligence,* SG24-6340.
- *Business Process Management: Modeling through Monitoring Using WebSphere V6 Products,* SG24-7148.

Other publications

These publications are also relevant as further information sources:

- A white paper, *Enabling Generic Web Services Interfaces for Business Process Choreographer,* by Eric Erpenbach and Anh-Khoa Phan.
- *“IBM Information On Demand: delivering the business value of information”, February, 2006, G299-1918.*
- IBM Business Consulting Services publication, *“The Agile CFO”, G510-6236,* which you can access at:

  `http://www-1.ibm.com/services/us/bcs/html/bcs_landing_cfostudy.html?ca=bv&tactic=105AS02W`

Online resources

These Web sites and URLs are also relevant as further information sources:

- WebSphere Portal Information Centers:

  `http://www-128.ibm.com/developerworks/websphere/zones/portal/proddoc.html`
- WebSphere Application Server Information Center:

  `http://www-306.ibm.com/software/webservers/appserv/was/library/`

- IBM service oriented architecture (SOA):

  http://ibm.com/soa/

- IBM Information On Demand Center of Excellence:

  http://ibm.ascential.com/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks 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

C

D

IBM

Redbooks

Improving Business Performance Insight . . . With Business Intelligence and Business Process Management

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

Improving Business Performance Insight . . . .

with Business Intelligence and Business Process Management

Proactive management of business measurements and goals

Real-time enterprise business intelligence

Monitor and manage business processes

In this IBM Redbook, we describe and demonstrate how to implement enterprise performance insight. This is an initiative that has a primary focus on the integration of Business Process Management and Business Intelligence. With this capability, management has an enterprise-wide view of their business that can enable proactive business management. We discuss the techniques, architectures, and processes used to define and implement such an environment. Among the specific techniques and technologies used are key performance indicators, process alerts, management dashboards, analytic applications, application integration, process modeling and monitoring, and real-time business intelligence. The products featured are DB2 UDB, DB2 Alphablox, WebSphere Information Integrator, WebSphere Portal, WebSphere Business Monitor, and WebSphere Business Modeler.

Performance insight is an element of a more global initiative called business innovation and optimization (BIO). With this we can enable business performance management, an initiative for the effective use of people, processes, assets, and technology to proactively achieve business goals and measurements. It enables strategic alignment of business and technology, resulting in real-time access to data and continuous process and data flow, for proactive business management, and business goal attainment.

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

SG24-7210-00 ISBN 0738494755