IBM

Implementing Technology to Support SOA Governance and Management

Guidelines for establishing SOA governance and management

Service lifecycle management with Rational Asset Manager

Runtime discovery with WebSphere Business Services Fabric

Martin Keen
Dirk Adamski
Indradri Basu
Peter Chilcott
Mark Eames
Mark Endrei
Bernardo Fagalde
Richard Raszka
S. Derek Seabury

Redbooks

ibm.com/redbooks

International Technical Support Organization

Implementing Technology to Support SOA Governance and Management

December 2007

SG24-7538-00

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

First Edition (December 2007)

This edition applies to Rational Asset Manager V6.0, WebSphere Business Services Fabric V6.0.2, and WebSphere Service Registry and Repository V6.0.2.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

Redbooks (logo) ®
developerWorks®
z/OS®
AIX®
Build Forge®
Candle®
ClearCase®
ClearQuest®
Cloudscape™
CICS®
DataPower®
Domino®
DB2 Universal Database™
DB2®
IBM®
IMS™
Lotus Notes®
Lotus®
NetView®
Notes®
OMEGAMON®
Rational Unified Process®
Rational®
Redbooks®
RUP®
System z™
Tivoli Enterprise Console®
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

IT Infrastructure Library, IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

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

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

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

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

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

Preface

Implementing service-oriented architecture (SOA) governance and management requires the consideration of three pillars: people, process, and technology. This IBM® Redbooks® publication addresses the third pillar, technology, demonstrating how to build advanced SOA solutions.

The first part of this book introduces SOA governance and management. It defines the IBM SOA governance and management method of plan, define, enable, and measure. This part also addresses the IBM SOA Foundation, SOA metadata, and the service life cycle.

The second part of the book uses a fictional telecommunications company business scenario to illustrate the use of technology in implementing SOA solutions with governance and management. This includes a detailed hands-on introduction to Rational® Asset Manager, highlighting its governance capabilities. An additional chapter discusses the use of WebSphere® Business Services Fabric as a process runtime and WebSphere Service Registry and Repository as a runtime repository for service discovery.

We discuss how a layered mediation architecture supports SOA governance and how DataXtend Semantic Integrator can be used to implement service components for data mediation. Finally, we discuss additional solutions for SOA management.

Target audience for this book

This book includes topics for a foundational understanding of the concepts, challenges, principles, and the value proposition of SOA governance and management. More particularly, the book focuses on:

- The technologies and their roles in the support of SOA and SOA governance
- How these technologies support SOA and SOA governance activities

This book is for Technology professionals, Program Managers, Business Analysts, and Business Sponsors, including C-level executives, either directly responsible for or wanting to understand more about:

- The role of governance and management in realizing benefits of SOA
- Technologies that support SOA and SOA governance and management

The team that wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Martin Keen is a Senior IT Specialist at the ITSO, Raleigh Center. He writes extensively about WebSphere products, SOA, and Patterns for e-business. He also teaches IBM classes worldwide about WebSphere, SOA, and Enterprise Service Bus (ESB). Before joining the ITSO, Martin worked in the EMEA WebSphere Lab Services team in Hursley, UK. Martin holds a bachelor's degree in Computer Studies from Southampton Institute of Higher Education.

Dirk Adamski is a Senior IT Architect at the IBM Software Group in Germany. He has 15 years of experience in IT business and is working as a Software IT Architect in the Finance Services Sector. Before working in this role, Dirk was a member of the Tivoli® Services team and earlier in the IBM Global Services unit. He has experience in SOA and Systems Management. Dirk is a graduate electrical engineer.

Indradri Basu is a Senior Software Engineer in India Software and Technology Lab. He has nine years of experience in Information Technology and Systems Engineering across various domains. He hold a bachelor's degree of science. His areas of expertise include Rational tools administration, Configuration Management, Architecting rational tools deployment, and Integration.

Peter Chilcott is a Senior IT Architect with the IBM GBS Application Services Architecture Practice in Melbourne, Australia. He has 22 years of experience in the IT industry working primarily in the software engineering, application integration, and performance domains. His areas of expertise include SOA, technical architecture, and J2EE™ application servers.

Mark Eames is a Technology Architect with Telstra Corporation Limited, in Australia. He has many years of experience in Enterprise Information Technology and Software Engineering. He holds a master's degree in Technology from Royal Melbourne Institute of Technology. His areas of expertise include Distributed Computing and Enterprise Architecture.

Mark Endrei is an IT Architect with IBM Australia. He has eight years of experience in the fields of application integration and SOA. He holds a bachelor's degree in Computer Systems Engineering from the Royal Melbourne Institute of Technology and an MBA in Technology Management from Deakin University/APESMA.

Bernardo Fagalde is an IT Architect in Uruguay working for IBM GBS. He has three years of experience in the SOA and Integration field and also seven years working for IBM. He holds a bachelor's degree in Computer Engineering from the public university of Uruguay (*Universidad de la República - Facultad de Ingeniería - Ingeniero en Computación*). His areas of expertise include SOA, Enterprise Architecture, Integration, J2EE, and WebSphere family. He has co-authored other IBM Redbooks publications, such as *WebSphere Application Server V6.1: System Management and Configuration*, SG24-7304, and *WebSphere Application Server V6.1: Planning and Design*, SG24-7305.

Richard Raszka is an IT Architect with the GBS Architectural Practice in Melbourne, Australia. He has 22 years of experience in the IT industry working in software development. He holds a degree in Mechanical Engineering, Mathematical Sciences, and an MBA. His areas of expertise include J2EE, WebSphere Application Server, Rational Application Developer, and developing and solutioning SOA applications for clients. He has previously co-authored the IBM Redbooks publication, *Rational Application Developer V6 Programming Guide*, SG24-6449.

S. Derek Seabury consults as a Systems Architect for Progress Software in Boston, Massachusetts. He has 14 years of experience in IT and degrees in Computer Science and Electrical and Computer Engineering from the Johns Hopkins University. His areas of expertise include Common Model Architecture, SOA design, managing SOA development, DataXtend Semantic Integrator, and the NGOSS SID model. Derek is active in the TM Forum in the development of standards for system interaction and data exchange.

Figure 1 Indradri, Dirk, Bernardo, Mark Eames, Derek, Peter, Richard, Martin, and Mark Endrei

A special thanks to our sponsors for their enlightened input and leadership:

Henry Co
Software IT Architect, Telstra, IBM Software Group A/NZ

Peter Bittner
Client IT Architect, Telstra Integrated Account, IBM Australia

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

Jane Chen
IT Architecture Profession Leader, Global Business Services, IBM Australia

Barbara McKee
Architect SOA Foundation, IBM Software Group, IBM US

William A Brown
Executive Architect SOA Center of Excellence, EA&T, IBM US

Grant Larsen
STSM, Chief Architect - Asset Management, Rational software, IBM US

William Tegan
Associate Partner, Banking Sector, IBM US

Garry Moore
Senior IT Architect, IBM Canada

Kerrie Holley
IBM Fellow and WW CTO SOA Center of Excellence, IBM US

Dr Gili Mendel
Manager, Rational Asset Manager Development, IBM US

Chris Frost
IT Specialist, IBM Software Group, IBM Australia

Scott Nettleship
Project Manager, IBM Software Group Tivoli, IBM US

Sanjoy Banik
Senior Consultant, IBM Software Group WPLC, IBM Australia

Michael Brokmann
Senior IT Architect, IBM Software Group Tivoli, IBM Germany

Jon Richter
SOA Governance Lead, Software Group Services, IBM US

Nick Houghton
Integration Consultant, Global Business Services, IBM Australia

Davyd Norris
Senior Presales IT Architect, IBM Australia

Patricia Vuong
IT Specialist, Global Business Services, IBM Australia

Ganesh Nagasamy
Architecture Services, Global Business Services, IBM Australia

Naveen Balani, Rohit Sardesai, Chaitanya Laxminarayan, Alvin Richardson, Albert Császár
WebSphere Business Services Fabric Development, IBM Software Group

Martin Smithson
WebSphere Service Registry and Repository Development, IBM Software Group UK

Rishi S Balaji
Associate IT Architect, Global Business Services, IBM India

Devaprasad Nadgir
Solutions Architect - SOA Technology Practice, IBM India

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will 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 books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

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

ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@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

Part 1

Overview

1

Introduction to SOA governance and management

IBM is the thought leader in the area of SOA governance. The IBM SOA Governance and Management Method (SGMM) describes an SOA governance model, and the approach and assets through which to implement this way of governance.

In this chapter, we first provide definitions relevant to service-oriented architecture (SOA) governance and management. The remainder of the chapter introduces concepts necessary to understand the relevance and importance of SOA governance and management in realizing the benefits associated with SOA. These concepts are introduced as:

- Definitions of SOA, service, SOA governance and management, and its relationships to other types of governance and management
- Relevance of SOA governance and management, as defined by SGMM
- Overview of the SOA governance life cycle and methodology

1.1 SOA governance and management definitions

In this section, we define what is meant by the terms:

- Service-oriented architecture
- SOA governance, including definitions of related terms:
  - Governance
  - Corporate governance
  - IT governance

1.1.1 Definition of service-oriented architecture

Service-oriented defines a method of integrating business applications and processes as linked services.

Service-oriented architecture (SOA) is an architectural style particularly suitable for an enterprise IT architecture that leverages the principles of service orientation to achieve a tighter relationship between business needs and the IT systems that support those needs. SOA supports a natural way of thinking about the organizations we operate and the world in which we live. In our daily lives, we depend on services provided by other organizations, departments, and people. Booking an automobile in for a maintenance service or invoking a mechanic to repair a car's radiator are services that we rely on in our lives. We know very little about how the service is fulfilled or managed, but we do know how to engage or invoke the service. Very often our perception of the quality of that service is dependent on how little we need to know about the internal working of the service and the extent to which it meets our functional (quality of work) and nonfunctional (timeliness, reliability, and so forth) needs. Alternatively, specific definitions of SOA varying with the person's organizational role and context (business, architecture, implementation, and operational) are outlined here:

- From a business perspective, SOA defines a set of business services composed to capture the business design that the enterprise wants to expose internally, as well as to its customers and partners.
- From an architectural perspective, SOA is an architectural style that supports service orientation.
- From an implementation level, SOA is fulfilled using a standards-based infrastructure, programming model, and technologies, such as Web services.
- From an operational perspective, SOA includes a set of relationships and agreements between service consumers and providers that specify the quality of service, as well as reporting on the key business and IT metrics.

For more information about service-oriented architecture, see *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240.

1.1.2 Service definition

There is no industry agreed upon standard definition of a service in SOA. A service is representative of a repeatable business task. A business service, such as managing an order, can be made up of subprocesses, such as:

- Capture the order (input)
- Process the order (operate)
- Dispatch the order (output)

Using a top-down methodology, you identify a business process and then, within that process, the set of tasks that are performed. A reasonable claim is that the tasks in a business process are *services* and the business process is a *composition of services*.

Services have the potential to become more granular (for example, more discrete tasks) as the tasks within a composite service are in turn understood and potentially decomposed. For example, capture order can be a discrete business service, or it can be a composite business service made up of other services, such as capture the customer details.

The ability for services to become more and less granular is only one important consideration in SOA governance and management. There are few agreed upon industry standards or principles that can be applied to validate a service. Compliance with some of the generally agreed upon industry principles and technical standards provides a baseline opportunity to validate a service. Generally agreed upon service validation principles include:

- Loose coupling

  Loosely coupling between a service provider and consumers means minimizing the number of things that a consumer needs to know about the provider or vice versa.

  Where loose coupling is not achieved, a change made to any aspect of a service relationship (provider and consumer) will either result in a change to the consumer or the provider (or, possibly, both). If a change is made by any party (the consumer, provider, or mediating infrastructure) to any aspect of a service that is decoupled, there should be no need to make subsequent changes in the other parties. Loose coupling is an important service validation principle, because it:

  – Reduces support costs, the impact of changes in business processes, and IT applications, and enhances adaptability

  – Reduces dependencies between applications, for example, to improve availability and managability of user applications

  Location transparency is a technical aspect of loose coupling. The consumer does not require explicit knowledge of where the service is executed.

- Encapsulation

  Access to functions and data must be through a well-defined interface that forms a contract between the service provider and the service consumer. *Encapsulation* hides any data or behavior in the service implementation that is specific only to the internal working of the service and irrelevant to the service consumer.

- Stateless

  The expectation that a service provider remembers a transient state relevant to a particular consumer adds complexity to service providers. Avoidance of complexity promotes the opportunity to provide more readily scalable and highly available service providers at a lower unit cost.

- Technology neutral

  Consumers on any platform must be able to invoke services provided on any platform.

Services are the first class citizens of a service-oriented architecture. An organization will need to establish additional principles and standards to validate any service as being worthy of investment, development, and maintenance. The establishment, assessment, and maintenance of such principles and standards is a function of SOA governance.

For more information about services in context of SOA, consider:

- *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240
- *z/OS Technical Overview: WebSphere Process Server and WebSphere Enterprise Service Bus*, REDP-4196

Business Process Management

A *composite business service* is comprised of multiple service components, each of which exposes a service interface. *Business process management* is the enabling concept and technology for composite business services. Business process management, both in concept and execution, must combine business processes, information, and IT resources with an enterprise's people, information, technology, and processes.

For more information about composite business services and business process management, see *Patterns: SOA Foundation - Business Process Management Scenario*, SG24-7234.

1.1.3 Governance definitions

The IBM SGMM defines *governance* as the establishment of chains of responsibility, measurement, policies, control mechanisms, and communication:

- *Chains of responsibility* are the establishment and assignment of decision rights. Roles are defined, and associated with those roles are responsibilities. Chains of responsibility signifies the assignment of accountability.
- *Measurement* is how to measure the effectiveness of the governance that is put in place. What key performance metrics need to be defined? What key performance indicators need to contribute to the initial goal?
- *Policies* are used to prescribe management direction, to guide an organization to meet its stated business needs and objectives, to ensure that the business conforms to prescribed legislation, to demonstrate management commitment, and to clearly define responsibilities of a particular party within an organization.
- *Control mechanisms* are instruments that are put in place to make sure that everyone is doing what they are supposed to be doing. Control mechanisms ensure compliance with the policies that have been defined. Control mechanisms operate by assuring compliance at various compliance checkpoints.
- *Communication* is the glue of governance and a significant element. Parties must be informed to enable compliant behavior, which drives the importance of communication.

A key principle of any governance is that it makes expectations clear and makes it easier for people to do the right thing rather than the wrong thing.

SOA governance is necessarily defined in context, in other words, its relationship to other forms of governance. The forms of governance relevant to the definition of SOA governance are:

- Corporate governance
- IT governance

Figure 1-1 on page 8 provides a view of the relationships among the forms of governance introduced in this chapter.

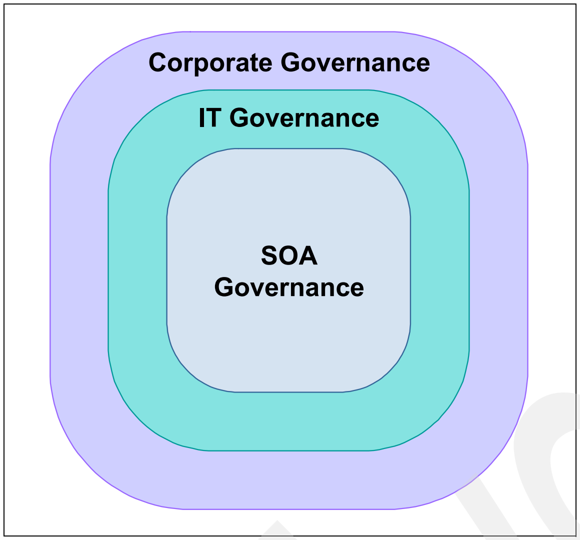


Figure 1-1 Governance relationships

Corporate governance

Corporate, or enterprise governance, establishes the rules and the manner in which an enterprise conducts business, based upon its strategy, marketplace, and principles of doing business. It defines for employees and for business associates the processes that are used to conduct operations and the manner in which people interact.

Beginning with the board of directors and extending throughout the organization, there are many aspects and levels of corporate governance. All aspects of the business are touched in some manner. Governance is applied to major functional areas of an organization. Organizations govern their financial assets, human resources, customer relations, intellectual property portfolio, and their Information Technology.

IT governance

IT governance is broader than SOA governance and refers specifically to the aspects of governance that pertain to an organization's information technology processes and the way that those processes support the goals of the business. Given the horizontal nature of IT, where almost everyone in the enterprise uses IT assets to complete their responsibilities, IT governance represents a significant part of corporate governance. IT governance as a subset of corporate governance deals with the management and control of IT assets, people, processes and infrastructures, as well as the manner in which the assets are managed and procured. IT governance also helps to define roles and responsibilities and to specify the decision rights and the accountabilities that will help to encourage desirable behavior in IT departments and establish accountability for the use of IT assets. IT governance also helps to define and realize best practices. For more information about the assignment of decision rights and measures for IT processes, consider those defined by Control Objectives for Information and related Technology (COBIT):

http://www.itgi.org

SOA governance

SOA governance identifies the changes to IT governance to ensure that the concepts and principles for service orientation and its distributed architecture are managed appropriately and that services are able to deliver on the business goals.

SOA governance addresses how an organization's IT governance decision rights, policies, processes, and measures need to be modified and augmented for the successful adoption of SOA, thus forming an effective SOA governance model. SOA governance provides an essential framework for achieving functional and nonfunctional (security, reliability, and performance) interoperability of services across line of business boundaries.

Because of its cross-functional and organizational aspects, SOA governance also provides a framework for examining several items necessary to manage services as another type of IT asset, such as:

- Maturity of service orientation within the enterprise
- Infrastructure enhancements for managing the usage of services in areas of security, monitoring, performance, versioning, and shared usage
- Enhancements to IT processes to address funding, sharing, incentives for sharing, and reuse of services, as well as for the identification, design, and specification of services
- Education and training
- Roles and responsibilities
- Organizational changes

SOA governance enhances IT governance by assigning decision rights, policies, and measures around the services, processes, and life cycle of SOA to address such concerns as:

- Service registration
- Service versioning
- Service ownership
- Service funding

- Service monitoring
- Service auditing
- Service diagnostics
- Service identification
- Service modeling
- Service publishing
- Service discovery
- Service development
- Service consumption
- Service provisioning
- Access to services
- Deployment of services and composite applications
- Security for services

1.1.4 Definition of SOA management

SOA management includes solutions and software for the effective management of production SOA services. SOA composite applications traverse multiple systems, platforms (Web servers, middleware, and databases), and other resource components of IT infrastructure. SOA management provides the capabilities required to efficiently manage SOA-based applications. The ability to monitor, diagnose, and control end-to-end implementations of SOA composite-based services and applications is a key component to deriving value from SOA implementations.

1.2 Relevance of SOA governance and management

SOA is a new approach to conceptualizing and building IT systems. SOA offers support of business functions and processes in a manner that is more agile and is more readily aligned to the needs of the business. Much of the value proposition associated with SOA emphasizes notions of reuse, loose coupling, and cross-organizational, cross line of business sourcing of components (services). Realizing the SOA value proposition has implications on the mechanisms by which the various IT processes are to be governed.

A major challenge in adopting an SOA is that many groups, both internal and external to the organization, contribute to the execution of business processes. With an SOA, once-siloed information, data, and function are now exposed as services. These services can be potentially shared across departments, lines of business, and even organizational boundaries. The distribution of information, data, and function across departmental and organizational boundaries raises concerns about decision rights and process compliance, measurement, and control. For SOA to deliver business value and to remain consistently and predictably aligned to realizing both the business and the SOA vision, key questions in relation to decision rights, process measurement, and control need to be able to answered. Such questions include:

- Who makes the decision and what criteria do they use to decide whether a service can be accessible to other applications?
- Who funds, owns, and operates shared services?
- What specification and implementation standards are applied across the organization to assure security and interoperability?
- How does the enterprise determine whether a service achieves expected results?
- How will changes to services be managed?
- Who is responsible for and how are the end-to-end availability and performance for composite business functions measured, managed, and assured?

1.3 Relationship between IT and SOA governance

SOA governance has a specialization relationship with IT governance. Enterprises that have existing IT governance models in place will generally have an easier time establishing an effective SOA governance model to govern the life cycle of a service and the life cycle of the portfolio of services. Even in the presence of absent or incomplete IT governance, SOA governance provides an opportunity to revisit shortcomings in the IT governance domain and serves as a catalyst to establish a stronger overall IT governance.

Table 1-1 on page 12 shows the specialization relationship between IT and SOA governance decision framework.

Table 1-1 Shows the relationship between IT governance and SOA decision categories

IT governance decisions	SOA governance decisions
IT vision	SOA vision
IT principles	SOA business and it principles
Business application needs	Service portfolio needs
IT architectural decisions	SOA architectural decisions
IT infrastructure standardization	SOA infrastructure standardization
IT investment and prioritization	SOA service candidate investment and priority

These key categories of decisions of an IT governance model are shown on the left column:

- *IT vision* – Communicates the decisions related to the purpose and behaviors expected to achieve that purpose within the IT organization within the context of the organizational vision.
- *IT principles* – Are used to guide consistent decision making aligned with the purpose of the IT organization. Principles provide the rationale, objectives, and implications of the principle. IT principles need to be reflective of the high-level strategic business principles that are defined by an enterprise.
- *Business application needs* – These decisions focus on identifying what business functionality is required to realize strategic business objectives and which applications will be leveraged or procured to provide that business functionality.
- *IT architectural decisions* – Encompasses decisions consistent with IT principles to both define and standardize the relationships among information, data, applications, and infrastructure. These decisions are usually communicated in a set of policies, standards, and best practices.
- *IT infrastructure standardization* – Provides and communicates the decisions necessary to operate and connect multiple applications, so that they can be managed, maintained, and integrated. These decisions determine common platforms and how to manage those platforms to ensure operability and interoperability.
- *IT investment and prioritization* – These decisions include the allocation of resources, to what the resources should be allocated, and how to reconcile the needs of various constituencies through the alignment of investment with the highest strategic priorities.

SOA governance also has to address these decision categories from an SOA perspective. These IT governance decision categories will manifest themselves in a slightly different and more specific form when considered from an SOA point of view.

From an SOA point of view, the decision categories become SOA governance decisions:

- *SOA vision* – Establishes the purpose of SOA in the context of both the organizational and IT vision.
- *SOA business and IT principles* – Core principles that define the SOA initiative, and all decisions going forward must be made in a manner that enables realization.
- *Service portfolio needs* – Effective service portfolio management is the key to the success of any SOA. An SOA-enabled IT must provide services that represent the functionality that the business needs to achieve the company's goals. How are these needs and services identified? How are they reflected in the portfolio? Who is ensuring that this is being done? How is the effectiveness of what has been implemented being measured to demonstrate that these needs are being met?
- *SOA architectural decisions* – The SOA value proposition is dependent on the ability for service providers and consumers to interoperate. Increased importance is placed on the establishment of standards and best practices to help define the rules of engagement for these parties so that functionality can be effectively shared and used by alternate consumers over time. What standards will service providers and consumers need to support to ensure interoperability? What are the best practices and guidelines to ensure high quality services that are usable across line of business and organizational boundaries?
- *SOA infrastructure standardization* – Shared infrastructure is critical to an SOA. SOA infrastructure, including runtime and design time service repositories and service bus, has a potential demand that will cross line of business or departmental boundaries. Determination of how that infrastructure will be funded, implemented, and managed needs to be understood and agreed upon. How will shared SOA infrastructure be procured and maintained? Which organizational group will fund it and who is responsible for maintaining it? These are critical issues that need to be resolved when implementing a successful SOA.
- *Service candidate investment and priority* – The business needs to commit to the fact that the service is potentially needed before a service is designed and subsequently developed, tested, and deployed. The business does this by committing funding for the service. However, there is an opportunity cost that is associated with the funding of this service and if one service is funded, it is

likely that other service candidates might have to wait. So, it needs to be decided how to determine whether a service is funded or not? What type of analysis must be done before funding is allocated? How does the enterprise prioritize service candidates to take advantage of the capital that is available and minimize the opportunity cost of funding one service for the sake of another? What incentives are there for a service provider to create high quality services and a consumer to use those services? These are probably the difficult and important questions to answer when implementing an SOA of value.

1.4 Drivers of SOA governance

We have established an understanding of what SOA governance is and its relationship to other forms of governance. We next address why SOA governance is important. What are the key drivers for consideration and realization of an SOA governance strategy? The key drivers discussed here are:

- Realize SOA business benefits

  SOA governance provides a centralized, planned, measured, and formalized approach to delivering service-oriented functionality in order to enable interoperability between potential service consumers and providers. This approach also allows the company to realize economies of scale, and as the approach matures, the resources become more efficient in delivering this functionality, allowing a quicker time to market.

- Reduce the risk of SOA implementation

  Potential service consumers need to access, understand, and accept the functionality that service providers deliver as a dependency. If this environment is not governed, the service consumers are forced to relinquish control of their destiny into the hands of the providers, without any guarantee from the providers that the functionality will behave, perform, or exist according to expectations. Alternatively, where service consumers are not able to locate or accept the provider’s functionality or behavior, the risk exists of a plethora of similar but different services with their associated life cycle costs. Governance allows us to define these expectations via Service Level Agreements (SLAs) in a binding contract and enforce the criteria outlined within the contract.

- Promote team effectiveness

  Communication is one of the key governance processes. Given SOA’s distributed and cross line of business environment, it is critical that a provider service in one part of the organization is usable by consumers in other parts of the organization. To enable this interoperability, the organization needs to work to the same corporate and IT principles, standards, and controls. SOA governance provides those communication mechanisms to improve effectiveness by leveraging governance processes and artifacts defined once and used across the organization.

1.5 Drivers of SOA management

With an understanding of the definition of SOA management (1.1.4, “Definition of SOA management” on page 10), we can now address the key drivers of an SOA management strategy. The key drivers or capabilities identified here are:

- Understand the relationship of services

  How does an organization understand, discover, and maintain the relationships between service consumers and providers for composite applications? Additionally, how does IT operations understand, discover, and maintain the relationships between services and the IT resources that support them?

- Manage services as resources

  To achieve the quality of service (QoS), availability, and performance that are required by the business, each service endpoint must be managed as a resource. Managed services must have real-time availability and performance metrics and a defined service level agreement. Who in the organization is responsible for this management and how do they fulfill their responsibilities?

- Identify the resources to manage

  A key challenge of SOA management is knowing how to identify which resources need to be managed.

- Monitor the end-to-end view in an integrated console

  For composite services that traverse lines of business or even organizational boundaries, who is responsible for the end-to-end QoS, and again, how do they manage and report compliance?

1.6 SOA governance life cycle

SOA governance must optimize the value of IT in achieving business outcomes. The IBM SGMM defines the SOA governance and management life cycle as shown in Figure 1-2 on page 16. This life cycle consists of four phases: plan, define, enable, and measure.

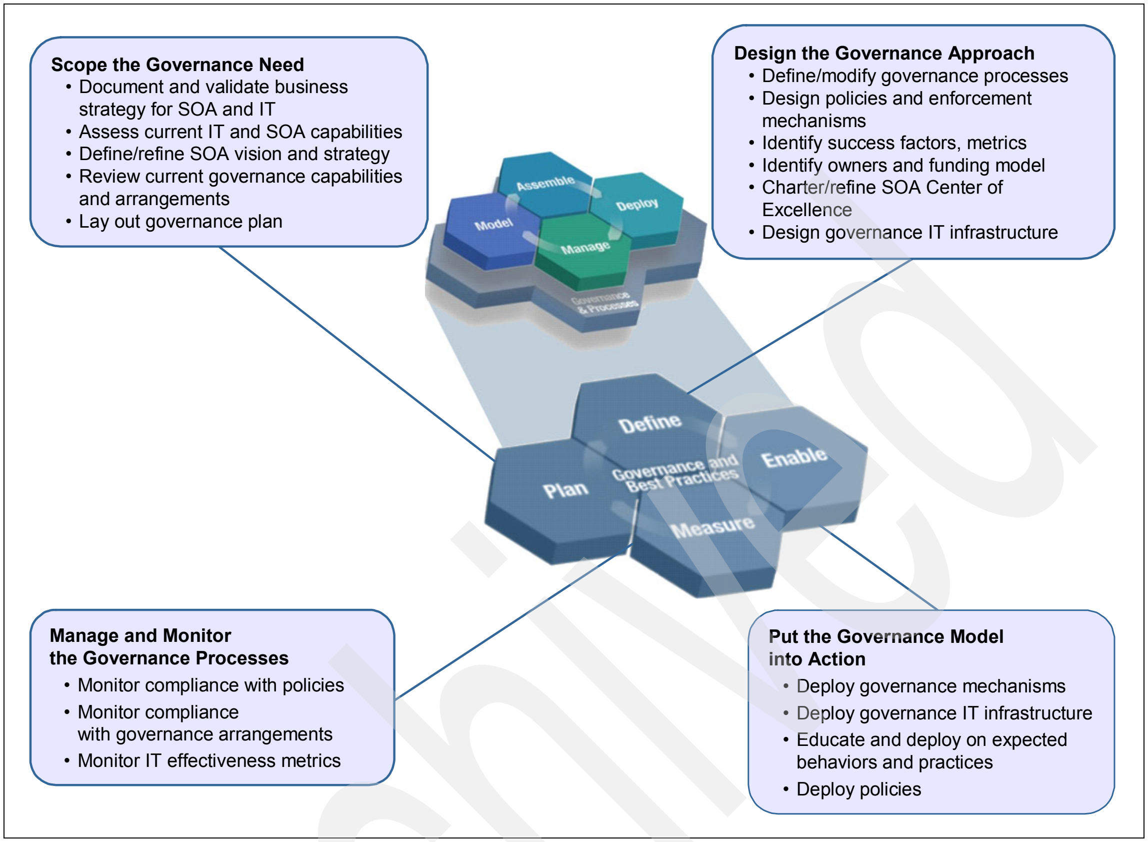


Figure 1-2 Four phases of the SGMM life cycle

Figure 1-2 shows that the four stages of the SGMM governance life cycle support the SOA Foundation life cycle of model, assemble, deploy, and manage. SOA governance effectively governs the key processes across the entire SOA Foundation life cycle.

SGMM is an approach for performing SOA governance. SGMM supports the four phases of the SOA governance life cycle of:

- Plan
- Define
- Enable
- Measure

SOA governance should leverage existing functional governance mechanisms. SGMM is flexible and can be adapted to alternate specific goals and extend existing governance mechanisms. Finally, SGMM is an iterative process providing opportunities to address immediate governance focus areas in an initial iteration and subsequent focus areas in subsequent iterations.

Figure 1-3 provides an overview of SGMM.

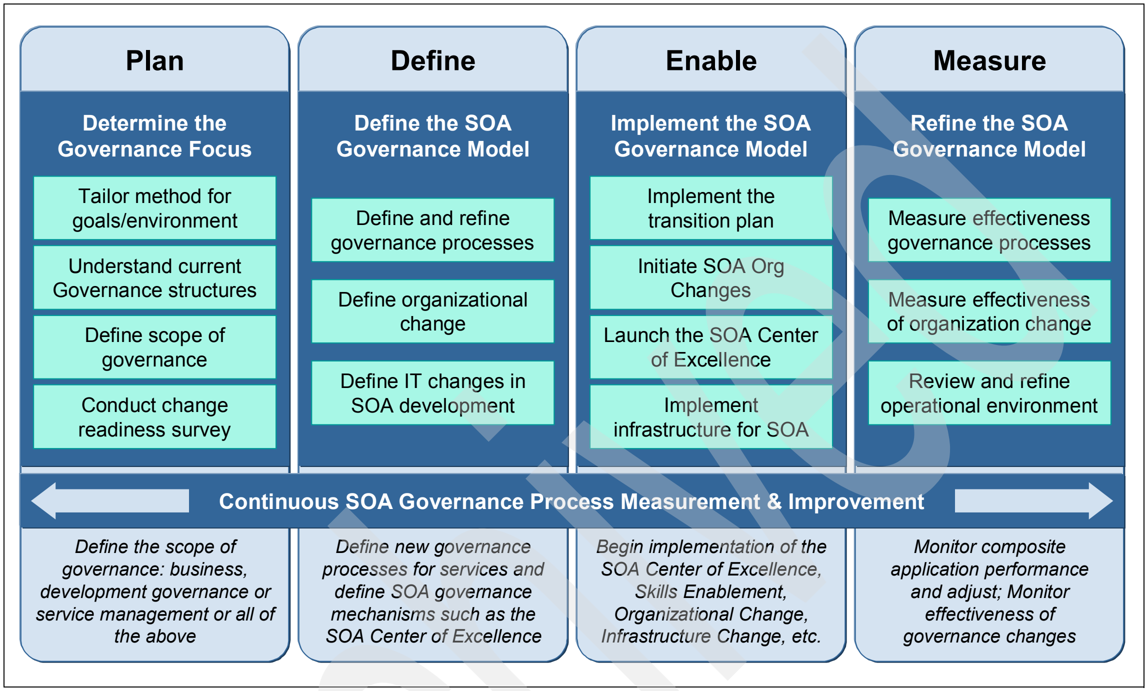


Figure 1-3 Summary of SGMM phases and activities

1.6.1 Plan

The planning phase of building an SOA governance framework focuses on understanding the overall scope of the governance opportunity within the organization and identifying areas for improvement. The main goals of the plan phase are to:

- Establish the need for governance and determine what the governance efforts need to be prioritized for an iteration of governance work.
- Commit to a strategy for SOA in the context of the overall business goals and IT strategy.
- Explicitly determine the level of IT and SOA capabilities.
- Articulate and refine the vision and strategy for SOA.

- Examine the existing governance environment and create a baseline for it. Plan to leverage and extend existing effective governance mechanisms.
- Define and refine the scope of the SOA governance model.
- Examine the organization's readiness to accept and adopt changes required to support SOA.

Most of these activities are people-centric and involve extensive communication and optimized collaboration.

1.6.2 Define

When the opportunities for improved governance are identified, business and IT people can work together to define and modify the current governance arrangements and mechanisms. New approaches to creating policies need to agreed on at this time. Other important governance decisions and mechanisms created during this phase can include:

- Define or refine SOA business, IT vision, and principles.
- Define and refine policies, standards, and quality gates and decision matrixes for the development and operational aspects of the service life cycle, including reuse, security, and operational management (change and version)
- Define governance mechanisms, including organizational structures (Architectural review board, executive steering committee, or Center of Excellence) and associated roles and responsibilities required to support the governance model.
- Identify any additional capabilities required, such as upgrades to the IT infrastructure.
- Identify needed skills and conduct staff training and mentoring on an ongoing basis.
- Agree on policies for service life cycle management across lines of business, including ownership, funding, and accountability models.
- Establish mechanisms to guarantee service levels.
- Define the infrastructure and tooling required to support SOA and SOA governance and management.
- Define measures and metrics that indicate the effectiveness of the governance model, including the definition of collection mechanisms and recipients.

These mechanisms and SOA governance decisions can accelerate the process of translating business design into IT design during the assembly phase of an SOA project.

1.6.3 Enable

Solutions to governance needs are put into action during this phase of establishing the SOA governance framework. Using the transition plan defined in the previous phase, the SGMM implements the various elements defined in the governance model. These solutions can include the deployment of new or enhanced governance arrangements. It is likely that communication mechanisms and education mechanisms will be rolled out to entrench the new governance arrangements within both the business and the IT decision-making communities. Governance activities within this phase influence how SOAs are deployed by enabling the policy enforcement infrastructure. Enable goals include:

- Enable the SOA governance organizational changes.
- Initiate and enable the SOA governance communication, education, and mentor plans.
- Monitor and refine the enablement plan.
- Deploy the infrastructure and tooling required to support the SOA governance and management plan.

1.6.4 Measure

During this phase, governance arrangements and mechanisms that were identified in the Define phase and deployed in the Enable phase are monitored. Activities occurring in this phase help ensure that processes, policies, and standards are being complied with and that the goals of the new governance framework are in fact being subsequently realized. If not, there is an opportunity for the business to refine and enhance its governance effectiveness by initiating a new cycle to enhance the SOA governance framework.

2

Establishing an SOA governance and management capability

This chapter provides recommendations, considerations, and guidelines for establishing a service-oriented architecture (SOA) governance capability within an organization. In this chapter, those recommendations, considerations, and guidelines are discussed as part of the IBM SOA Governance and Management Method (SGMM). As we address each SOA governance phase, we describe:

- The key activities of that phase
- Identification of the roles that are either responsible, accountable, consulted, or informed in relation to the activity
- The inputs and outputs of the activity

We additionally identify the roles and associated skills necessary to initially create and refine an SOA governance capability.

The presentation of the governance phases assumes the decision to establish an SOA governance capability needs to be funded, resourced, and deliverables or outcomes specified. We assume that the allocation of resources, review and acceptance of deliverables, and reporting of outcomes occur in the context of a broader executive group or governance steering committee.

2.1 Organizational roles

In this section, we identify a candidate set of roles required to establish an initial or refine an organizational SOA governance and management capability. In addition to describing the roles, we outline any specific skills for SOA governance. Any given role can be completed by one or more individuals. Alternatively, an individual can fulfill more than one role. There is no priority associated with the order in which the roles are introduced.

2.1.1 Business Analyst/Architect

The business analyst/architect is responsible for analyzing the goals and needs from a business perspective. They define requirements representing current and future operational scenarios (business and operational processes, models, use cases, plans, and solutions). In addition, they work with the business contacts and the IT Architect to ensure the proper translation of business requirements to IT solution requirements.

Skills

The Business Analyst/Architect skills include:

- Understand SOA

  Understand significant differences between SOA and other architectural patterns. Understand business needs that can be addressed through SOA.

- Business modeling

  Be able to document process flows and how they are used.

- Apply data and process models to business solutions

  Be able to lead discussions with project stakeholders to elicit information required to build flows. Be able to understand the implications of changing a process model.

- Understand business aspects of governance

  Be able to describe processes associated with and the business value of corporate governance, IT governance, and SOA governance to project stakeholders.

- Interpersonal skills

  Be able to facilitate and lead workshops in order to understand business goals and needs. Be able to discuss business terms with technical representatives in a way that is meaningful to the technical representatives.

2.1.2 SOA Governance Architects

The SOA Governance Architect's capabilities are comprised of three roles. These roles collectively contribute the skills necessary for understanding the current and future realization of best practices, governance processes, and the operational environment. The roles identified in this section are:

- SOA Initiative Architect
- Process Architect
- Infrastructure Architect

SOA Initiative Architect

The SOA Initiative Architect is responsible for identifying services from a decomposition of business processes and ensuring that these services enable realization of the business goals and drivers. They define reference architectures and create component models. This role is responsible for performance, availability, and scalability of the applications. The architect also maintains the functional interface to the application infrastructure. As part of the fulfilment of this role, they perform critical evaluation and selection of the packages, software, and hardware components of the architecture.

In the context of the introduction of SOA Governance, this role is concerned with the processes, procedures, mechanisms, and organizations that will govern it and the processes that will be controlled. This role can be best performed by someone who is an enterprise level SOA Architect. An enterprise level SOA Architect has broad skills in this discipline, including the ability to bridge between business and technology.

Skills

The SOA Initiative Architect must:

- Understand SOA

  The person performing this role must appreciate the best practices associated with SOA, including policies and standards. This person must be able to exhibit in-depth skills in relation to SOA best practices and prioritize those best practices that offer the best opportunity to deliver value to the organization.

- Understand governance

  Although many architects will have a working knowledge of governance, the architect fulfilling this role must have more comprehensive knowledge involving both theoretical and practical exposure.

- Understand development life cycle

  This person must have general knowledge of the development life cycles used in the organization to be able to determine the impact of SOA on the processes involved. The person performing this role must be able to lead others in the resolution of issues and in the implementation of development methodologies.

- Understand operational aspects of service implementation

  The person performing this role should at a minimum be able to perform tasks in this domain under the guidance of a more experienced professional. This person performs roles for the tasks within the operational domain that are typically performed by the Infrastructure Architect.

- Possess proven interpersonal skills

  This person must be able to present information clearly and concisely to business and technical stakeholders.

Process Architect

The Process Architect is responsible for the integrity of all process and procedure definitions and documentation.

Skills

The Process Architect must:

- Understand the business process

  This person must have hands-on knowledge and understanding of the particular processes, as well as the ability to effectively acquire knowledge of the business processes that are under scrutiny.

- Understand modeling

  The person who performs this role needs to have in-depth knowledge of process modeling techniques, notations, and tools.

- Possess proven interpersonal skills

  This individual must have good communication skills, in both content and presentation, to help other team members understand the processes

Infrastructure Architect

The Infrastructure Architect is responsible for the design of the physical (or operational) aspect of a total system, line of business, or technology domain, including network management, application, server, and storage platforms, and data management (database and archive) solutions. They are concerned with designing the architecture to reach desired system qualities, including performance, scalability, availability, security, and maintainability. They are responsible for the infrastructure, hardware, software, middleware, interfaces, application development, application management tools, systems performance measurements, and the specification of security for the domain or system.

Skills

The Infrastructure Architect must have:

- Knowledge of infrastructure components

  The person or persons performing this role have in-depth knowledge of operating systems, networking, security, systems management, and other infrastructure functions. This person is responsible for performing the tasks associated with the operational aspects of governance.

- Interpersonal skills

  The person performing in this role is responsible for presenting concepts and issues to both business and IT audiences. Ideally, the person performing this role can conduct workshop sessions.

- Knowledge of governance

  A working knowledge of governance is required, particularly in relation to infrastructure and operational best practices, standards, and policies.

- Knowledge of SOA

  A working knowledge of SOA is required, especially the impact of SOA on infrastructure components. The impact of SOA on infrastructure components includes the infrastructure to support SOA and governance.

2.1.3 Organizational Change Manager

The Organizational Change Manager is responsible for understanding capabilities in business, operations, and technology and assessing the impact of changes to the organization. Change agents help organizations determine what to do and how to do it. They use fact-based strategy development skills and make the link to successful execution in the following four areas of strategy:

- Determining where and how to compete
- Designing and implementing successful operating models
- Building the change architecture and organizing the people who will be enablers of it
- Determining strategic possibilities and realities of technology

Skills

The Organizational Change Manager must have the following skills:

- Organization analysis and design

  This person must have in-depth skills in analysis and design.

- Interpersonal skills

  This individual must have the ability to understand, interpret, and communicate complex ideas and concepts to all levels, including executive management. This person must be able to understand the basis for resistance to change and negotiate successful change.

- Governance

  The person who performs this role must have in-depth knowledge of governance requirements, with special emphasis on the organizational aspects of governance.

2.2 IBM SOA governance and management method

SGMM provides a unified process to guide the realization and evolution of SOA governance processes. The unified governance and management process is divided into four SOA governance disciplines:

- Plan
- Define
- Enable
- Measure

An SOA governance and management plug-in for Rational Method Composer includes practical processes with knowledge assets and guidance to get team members productive more rapidly and to put customized SOA governance processes and work products into action. This plug-in is a container for SGMM.

The Rational Method Composer for SOA Governance Plug-in can be obtained from:

http://www.ibm.com/software/awdtools/rmc/features/index.html

2.3 Plan

Effective IT and SOA governance results in more complete and cost-effective alignment of the IT organization in meeting the needs of the business. In the Plan phase, the goal is to understand, capture, and document the needs and priorities of the business, along with the role of the IT organization in meeting those needs. The current effectiveness of the IT organization governance processes is also assessed and measured, and the gaps are identified. Governance measures are used to assess how well the IT organization is aligned with the business and how well the business needs are met. From this analysis, the SOA governance vision (distinct from the SOA vision) and strategy, as well as a realization plan, are documented.

The activities discussed in the planning phase include:

- Initiate project:
  - Request documentation.
  - Conduct methodology customizing workshop.
  - Conduct project kickoff.
- SOA business discovery:
  - Determine existing governance structures.
  - Identify SOA business principles.
  - Create IT governance baseline.
- Determine IT readiness:
  - Understand the current environment and create a baseline for it.
  - Measure and evaluate existing governance capabilities.
  - Determine SOA change readiness.

The activity overview shown in Table 2-1 on page 28 provides a dashboard outlining the key planning activities, the accountable organizational roles, and those roles that participate or augment the activity, as well as identifying the inputs and outputs of the activity. In the following section, each activity is described in more detail.

Table 2-1 Activity overview for Plan phase

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
Project startup	Request documentation	Project Manager (Accountable)	► Business strategy ► Business drivers ► Current IT environment ► Current organization description ► Job roles, responsibilities, and competencies ► Operational model ► Organization's approved governance and SOA tools ► Principles, policies, and guidelines ► SOA vision ► Standards	► Documentation reflecting document state of inputs ► Distribution of documents to SOA governance team
	Conduct method tailoring workshop	► SOA Initiative Architect (Accountable) ► Organizational Change Manager, Business Analyst, Project Manager (Participate, Consulted)	► Capability interview guide prototype ► Method tailoring workshop presentation ► Project plans ► SOA governance and management project work breakdown structure	► Capability interview guide ► Project plans ► SOA governance and management project work breakdown structure
	Conduct project kickoff	► Project Manager (Accountable)	SOA governance and management project work breakdown structure	SOA governance and management project work breakdown structure

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
SOA Business Discovery	Determine existing governance structures	► Organizational Change Manager (Accountable) ► Business Analyst, Infrastructure Architect, Project Manager, SOA Initiative Architect (Participate/ Consulted)	► Current IT assessment ► Current organization assessment ► Process definition ► Process identification	► Current IT assessment ► Current organization assessment
	Identify SOA business principles	► Organizational Change Manager (Accountable) ► Business Analyst (Consulted/ Participate)	Principles, policies, and guidelines	Identification of the need for additional SOA enabling principles, policies, and guidelines
	Create IT governance baseline	► Organizational Change Manager (Accountable) ► Business Analyst, Project Manager, SOA Initiative Architect (Participate/ Consulted)	► Current organization description ► Strategic SOA essentials ► Tactical SOA essentials	IT governance baseline

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
Determine IT Readiness	Understand current IT environment	► SOA Initiative Architect (Accountable), ► Organizational Change Manager (Participate/ Consulted)	► Current IT environment ► Operational model ► Principles, policies, and guidelines ► Standards	A summarized form of: ► Current IT environment ► Operational model ► Principles, policies, and guidelines ► Standards for future architecture planning
	Measure/ evaluate governance capabilities	Organizational Change Manager (Accountable)	Capability interview guide prototype	► Capability interview guide ► Project plans
	Determine SOA change readiness	Organizational Change Manager (Accountable)	Change readiness interview guide prototype	► Change readiness assessment results ► Change readiness interview guide ► Situational analysis and impact report
Exit Plan Phase	Identify plan phase exit criteria	Executive Steering Committee		

2.4 Define

In the Define phase, the detailed governance plan is put in place for the current cycle. In particular, the processes to be governed are specified and prioritized, and the decision rights, policies, and measures for these processes are defined. In preparation for the next phase, detailed deployment plans are set. In some cases, these plans can include specifying or updating the structure and staffing of the SOA Governance Center of Excellence.

The activities discussed in the define phase include:

- Refine SOA principles and standards:
  - Update SOA business principles.
  - Update SOA IT principles and standards.
- Define SOA governance framework:
  - Define service ownership model.
  - Establish SOA governance mechanisms.
  - Refine SOA governance processes.
  - Refine metrics.
  - Document SOA governance mechanisms.
  - Refine roles and responsibilities for SOA governance organization.
- Create SOA Center of Excellence (CoE):
  - Define and expand upon SOA CoE structure.
- Define development and operational aspects:
  - Define policies for service reuse, IT compliance, and security.
  - Define quality gates for specification, development, and operational processes.
- Define SOA governance tools and infrastructure:
  - Define the tools and infrastructure building blocks for SOA governance.
- Create plan:
  - Define SOA governance plans.
  - Define SOA governance transition plan.
- Exit define phase:
  - Identify exit criteria for the define phase.

The activity overview shown in Table 2-2 on page 32 provides a summary outlining the key definition activities, the accountable organizational roles, and those roles that participate or augment the activity. Additionally, the inputs and outputs of the activity are identified. In the following section, each activity is described in more detail.

Table 2-2 Define phase activity overview

<table>
<tr><th>Activity group</th><th>Activity</th><th>Organizational role and responsibility</th><th>Inputs</th><th>Outputs</th></tr>
<tr><td rowspan="2">Refine SOA business principles and standards</td><td>Update SOA business principles</td><td>► Organizational Change Manager (Accountable)
► Business Analyst (Participate/ Consulted)</td><td>Principles, policies, and guidelines</td><td>Refined principles, policies, and guidelines</td></tr>
<tr><td>Update SOA IT principles and standards</td><td>► SOA Initiative Architect (Accountable)
► Organizational Change Manager (Participate/ Consulted)</td><td>Principles, policies, and guidelines</td><td>Refined principles, policies, and guidelines</td></tr>
</table>

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
Create SOA governance framework	Define service ownership model	► SOA Initiative Architect (Accountable), ► Business Analyst, Infrastructure Architect, Organizational Change Manager (Participate/ Consulted)	► Component business model ► Current organization description ► Service model	Service ownership model
	Establish SOA governance mechanisms	► SOA Initiative Architect (Accountable) ► Organizational Change Manager, Process, Engineer, Project Manager (Participate/ Consulted)	► Current IT environment ► Current organization assessment ► Current organization description ► Future organization design ► Job roles, responsibilities, and competencies	► Future organization design ► Job roles, responsibilities, and competencies ► Process definition ► SOA governance mechanisms
	Refine governance processes	► SOA Initiative Architect (Accountable) ► Business Analyst, Organizational Change Manager, Process Engineer (Participate/ Consulted)	► Future organization design ► Process definition	► Future organization design ► Process definition ► Responsible/ Accountable/ Consult/ Informed (RACI) matrix
	Refine metrics	► Organizational Change Manager (Accountable) ► Process Architect, Business Analyst (Participate/ Consulted)	Process definitions	Measurement program

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
	Document SOA governance mechanisms	► SOA Initiative Architect (Accountable), ► Organizational Change Consultant, Process Engineer, Project Manager (Participate/ Consulted)	► Current IT environment ► Current organization assessment ► Current organization description ► Future organization design, job roles, responsibilities, and competencies	► Future organization design ► Job roles, responsibilities, and competencies ► Process definition ► SOA governance mechanisms
	Define roles and responsibilities for the SOA governance organization	► SOA Initiative Architect (Accountable), ► Organizational Change Consultant (Participate/ Consulted)	► Current organization assessment ► Current organization description	► Future organization design ► Job roles, responsibilities, and competencies ► RACI (Responsible/ Accountable/ Consult/ Informed) matrix

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
Create CoE	Create the SOA CoE Structure	► SOA Initiative Architect (Accountable), ► Organizational Change Manager, Project Manager (Particpate/ Consulted)	► Current IT environment ► Current organization assessment ► Current organization description ► Future organization design ► Job roles, responsibilities, and competencies	Future organization design
	Expand upon the SOA CoE	► SOA Initiative Architect (Accountable) ► Organizational Change Manager, Process Engineer, Project Manager (Participate/ Consulted)	► Current IT environment ► Current organization assessment ► Current organization description ► Future organization design ► Job roles, responsibilities, and competencies ► Process definition ► Process identification	► Architecture management ► Architecture management framework ► Future organization design ► Procedures documentation

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
Define specification, development, and operational aspects	Define policies for service reuse, IT compliance, and security	► SOA Initiative Architect (Accountable) ► Infrastructure Architect, Process Architect, Organizational Change Consultant (Participate/ Consulted)	► Current organization description ► Principles, policies, and guidelines ► Standards	► Principles, policies, and guidelines ► Standards
	Define quality gates for development process	► SOA Initiative Architect (Accountable) ► Infrastructure Architect, Process Architect, Organizational Change Manager (Participate/ Consulted)	Process definition	► Process defintion ► Quality gates checklist
	Define quality gates for operational process	► Infrastructure Architect (Accountable) ► Process Architect, Organizational Change Manager (Participate/ Consulted)	Process definition	► Process defintion ► Quality gates checklist
Define SOA governance tools and infrastructure	Define tools, infrastructure building blocks, and architecture for SOA governance	► Infrastructure Architect (Accountable) ► Process Architect, Organizational Change Manager (Participate/ Consulted)	► Current IT environment ► Operational model ► Standards	Key decision record

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
Create plan	Build SOA governance plans	► Project Manager (Accountable) ► Business Analyst, Infrastructure Architect, Organizational Change Manager, SOA Initiative Architect (Participate/ Consulted)	N/A	► Communications plan ► Education and training plan ► Mentoring plan
	Define SOA governance transition plan	► Project Manager (Accountable) ► Business Analyst, Infrastructure Architect, Organizational Change Manager, SOA Initiative Architect (Participate/ Consulted)	Project plans	Integrated transition plan
Exit review	Operational change readiness support	► Organizational Change Manager (Accountable) ► Business Analyst (Consulted/ Participate)	► Change readiness asessment ► Change readiness interview guide	► Change readiness assessment results ► Situational analysis and impact report
Exit plan phase	Review and endorse define phase milestone deliverable	Executive steering committee	► SOA governance transition plan ► Operational change readiness	Endorsement and resources to proceed to Enable

2.5 Enable

The Enable phase is when the defined solution is rolled out to the organization. In this phase, roles are assigned, staff are trained, the decision rights can be automated in workflow tools, and the metrics collection and report mechanisms are put in place.

The activities discussed in the enable phase are grouped under a single execution-oriented activity group, Execute enablement. These activities are:

- Execute a transition plan.
- Initiate SOA governance organizational changes.
- Initiate SOA CoE.
- Initiate SOA governance education plan.
- Initiate SOA governance mentoring plan.
- Initiate SOA governance communication plan.

The activity overview shown in Table 2-3 provides an overview outlining the key Enable phase activities, the organization roles accountable and those that participate or augment the activity. Additionally, the inputs and outputs of the activity are identified. In the section following each activity is described in more detail.

Table 2-3 Enable phase activity overview

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
Execute enablement	Execute transition plan	► Project Manager (Accountable) ► Organizational Change Consultant ► SOA Initiative Architect (Consult/ Participate)	► Integrated transition plan ► Project plans	► Integrated transition plan ► Project plans
	Initiate SOA governance organizational changes	► SOA Initiative Architect (Accountable) ► Organizational Change Manager, Business Analyst, Project Manager (Participate/ Consulted)	SOA governance and management project work breakdown structure	SOA governance and management project work breakdown structure

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
	Initiate SOA CoE	► SOA Initiative Architect (Accountable) ► Organizational Change Consultant, Process Engineer, Project Manager (Consult/ Participate)	► Architecture management ► Architecture management framework ► Current IT environment ► Current organization assessment ► Current organization description ► Future organization design ► Job roles, responsibilities, and competencies ► Process definition ► Process identification ► Project plans	► Architecture management ► Architecture management framework ► Future organization design ► Mentoring plan ► Procedures documentation ► Project plans
	Initiate SOA governance education plan	► Project Manager (Accountable) ► Organizational Change Manager, SOA Initiative Architect (Consult/ Participate)	Education and training plan	Education and training plan
		► SOA Initiative Architect (Accountable) ► Organizational Change Manager (Consult / Participate)	Mentoring plan	Mentoring plan

Activity group	Activity	Organizational role and responsibility	Inputs	Outputs
	Initiate SOA governance communi-cation plan	► Project Manager (Accountable) ► SOA intiative architect (Consult/ Participate) ► Organizational Change Manager (Consult/ Particpate)	Communications plan	Communications plan
	Update and refine the transition plan	► Project Manager (Accountable) ► SOA Initiative Architect (Consult/ Participate) ► Organizatinal Change Manager (Consult/ Participate) ►	► Integrated transition plan ► Project plans	► Integrated transition plan ► Project plans
	Implement tools and infrastructure for SOA governance	► Infrastrucute Architect	► Current IT environment ► Key decision record ► Standards	► SOA governance tools/ infrastructure implemented
Enable milestones	Enable milestones defined	N/A	N/A	N/A

2.6 Measure

In this phase, the governance approach is executed and tuned. The governance metrics — those that show alignment with the business — are gathered. These metrics are used in the next cycle to revise the governance approach.

The activities discussed in the measure phase are:

- Execute measurement:
  - Measure effectiveness of SOA governance process.
  - Measure effectiveness of organizational changes.
  - Review and refine the operational environment.
- Measure milestones:
  - Initiate the governance mentoring plan.

The activity overview shown in Table 2-4 on page 42 provides a dashboard outlining the key measure phase activities, the accountable organizational roles, and those roles that participate or augment the activity. Additionally, the inputs and outputs of the activity are identified. In the following section, each activity is described in more detail.

Table 2-4 Measure phase activity overview

Activity group	Activity	Organizational role and responsibility	Input	Output
Execute measurement	Measure effectiveness of SOA governance process	► Organization Change Manager (Accountable) ► Process Engineer, SOA Initiative Architect (Participate/ Consult)	► Measurement program ► Process definition	► Measurement program ► Process definition
	Measure effectiveness of organizational changes	► Organization Change Manager (Accountable) ► Process Engineer, SOA Initiative Architect (Participate/ Consult)	► Process definition ► Quality gates checklist	► Process definition ► Quality gates checklist
	Review and refine the operational environment	► Organization Change Manager (Accountable) ► Process Engineer, SOA Initiative Architect (Participate/ Consult)	► Process definition ► Quality gates checklist	► Process definition ► Quality gates checklist

2.7 SOA management

SOA management includes solutions for managing and monitoring composite applications[1]. We highlight three key points regarding SOA management.

First, it is important to understand the relationship of service consumers and providers for composite applications. SOA brings the benefits of application

[1] Defines an application built by combining multiple services. A composite application consists of functionality drawn from several different sources within an SOA.

reuse. The importance of monitoring and managing the availability and performance of the application functionality exposed as a service increases when reused by service consumers that depend on this functionality.

Second, composite applications span the various layers of the architecture. Composite applications drive a mind shift in the management approach. There is a need for monitoring and management tooling that cover the end-to-end view of the composite application, as well as provide detailed information about performance and availability metrics for the individual components.

Third, SOA management is used to ensure that the management of nonfunctional requirements is aligned with the business objectives. Monitoring and management of the composite application must include specific metrics to ensure that SLAs are meeting the business objectives.

Historically, business applications were built with a monolithic purpose (silos) and were typically inflexible. While this kind of architecture can be effective, it is often very difficult to change and integrate with other applications within the enterprise and between enterprises because custom-coded connections are required.

Within the SOA solution domain, business processes increasingly depend on integrated services that form multi-tier, composite applications. Composite applications often span architectural layers, including service consumers, business processes, services, service components, and operational systems.

SOA management includes solutions and software for managing and monitoring SOA composite applications and supporting infrastructure across architectural layers. In this section, we highlight the key challenges for SOA management:

- Understand the relationship of the services.
- Manage services as resources.
- Ensure nonfunctional requirements are achieved.
- Identify the resources to manage.
- Monitor the end-to-end view in an integrated console.

2.7.1 Understand the relationship of the services

It is important to understand the relationship between service consumers and providers for composite applications. SOA brings the benefits of application reuse. Although reuse of existing applications as services has many benefits, you must consider the management implications. The importance of monitoring and managing the availability and performance of the application functionality exposed as a service increases when the service is reused by service consumers that depend on this functionality.

For example, in a siloed approach each application can have its own tax calculation functionality. In an SOA composite application, the tax calculation can be exposed and consumed as a common service. This common service has the benefits of flexibility, reuse, cost savings, and so on, but also has increased dependency and must be monitored and managed accordingly.

2.7.2 Manage services as resources

To achieve the quality of service (QoS) defined by the business, each service endpoint should be managed as a resource. This includes the invocation of services (service consumer), as well as the application functionality exposed as a service (service provider). When speaking in general terms, management of services can refer to a range of services technologies. In the context of our example scenario, services are implemented as Web services.

Managed services must have real-time availability and performance metrics and a defined service level agreement. Like other resources, services are deployed, configured, versioned, monitored, managed, secured, and audited. When down, the management tooling must provide a means of troubleshooting and, better still, a method of monitoring and alerting of issues before failure.

2.7.3 Ensure nonfunctional requirements are achieved

SOA management is used to ensure nonfunctional requirements of the IT architecture are aligned with the business objectives. Monitoring and management of the composite application must include specific metrics to ensure SLAs are meeting the business objectives. The management solution must also consider how to report on adherence to the SLAs.

2.7.4 Identify the resources to manage

A key challenge of SOA management is knowing how to identify what resources to manage. Knowing what should be managed is derived from analyzing the nonfunctional requirements and SLAs. The identification of resources to be managed is performed at the time of the solution analysis and design and by discovery in the run-time environment.

2.7.5 Monitor the end-to-end view in an integrated console

In a siloed application architecture, resources are often managed by separate operators or specialists, in separate management consoles.

Composite applications require a mind shift in the management approach. You need monitoring and management tooling that cover the end-to-end view of the composite application, as well as provide detailed information about performance and availability metrics for the individual resources. Also, the *operator* must have an end-to-end monitoring role.

There are several perspectives to consider for the end-to-end view. The end-to-end view includes both horizontal and vertical views. The horizontal view is the view of the transaction, for example, Web browser → Web service client → Web service → CICS® Transaction Server. The vertical view is the view of the service invocation through the architectural abstraction layers, including consumer, business process, composite application, (Web) service endpoints, concrete application artifacts that support the service (such as mediation primitives), and operational environment.

2.8 Methodologies and best practices

In this section, the methodologies *IBM Rational Unified Process®* and the *IBM Tivoli Unified Process* are described.

IBM has built these methodologies based on standards and best practices to enable the realization of the SOA life cycle. By using these methodologies, it is possible to combine the development and the operation of services and composite applications.

Every time the SOA life cycle is iterated, it leads to the generation of better, more refined services.

2.8.1 IBM Rational Unified Process

IBM Rational Unified Process, or RUP®, is a configurable software development process platform that delivers proven best practices and architecture that enable you to select and deploy only the process components that you need for your projects. With the industry-proven best practices of the Rational Unified Process methodology at its core, the RUP platform includes:

- Tools to configure RUP for your project's specific needs
- Tools to develop your own internal knowledge into process components
- Powerful and customizable Web-based deployment tools
- Online community for exchanging best practices with peers and industry leaders

For more information use the following sources:

- *IBM Rational Unified Process* is available at:

  http://www.ibm.com/software/awdtools/rup/

- *The IBM Rational Unified Process for System z*, SG24-7362
- *Using a Single Business Pattern with the Rational Unified Process (RUP)*, REDP-3877
- *Building SOA Solutions Using the Rational SDP*, SG24-7356

2.8.2 IBM Tivoli Unified Process

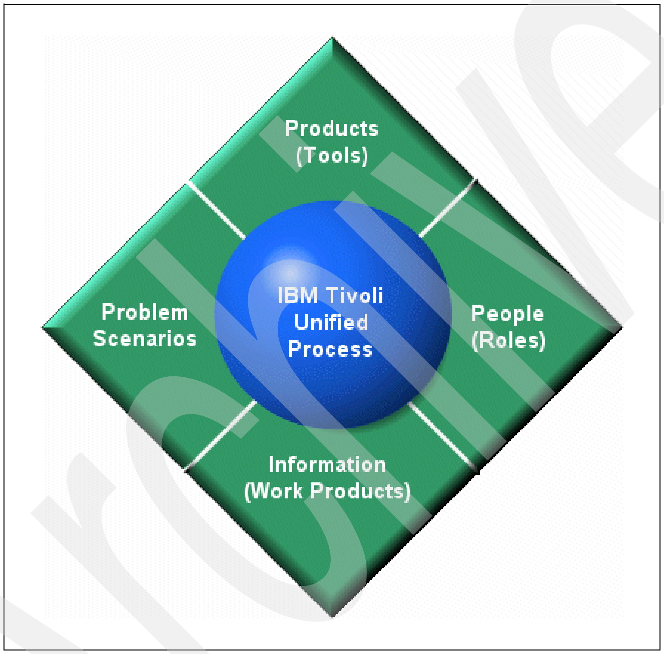


Figure 2-1 IBM Tivoli Unified Process

The IBM Tivoli Unified Process (ITUP) (Figure 2-1) delivers a prescriptive approach to IT Service Management. It is a Web-based process navigation tool, which describes IT processes. It details how IT Service Management (ITSM) can be achieved through the use of current management applications and the integration of new process-oriented products. The IBM Tivoli Unified Process complements the IBM Rational Unified Process with the focus on building applications and business services with inherently designed manageability for true IT life cycle support.

The following points are new in the current version ITUP V2.1.4:

- IBM Process Reference Model for IT (PRM-IT) is included in the ITUP Tool. PRM-IT is a process reference model that covers all 41 processes under the control of the Chief Information Officer.
- Operational processes covered by ITIL® best practices have been documented down to the granular task level.
- The current version includes a detailed mapping from the ITIL Version 3 processes and practices to the ITUP processes.

IBM Tivoli Unified Process (free download) is available at:

http://www.ibm.com/software/tivoli/governance/servicemanagement/itup/tool.html/

Standards for SOA management

IBM supports merging Service Management practices, controls, standards, and language into the IBM Tivoli Unified Process (2.8.2, "IBM Tivoli Unified Process" on page 46). A free downloadable version of the Process Reference Model for IT shows how to bring together best practices, controls, and standards from ITIL, ISO, COBIT, eTOM, CMMI, and eSCM.

We list these standards in more detail in the following sections.

ITIL

IT Infrastructure Library® (ITIL) is a best practice collection for IT operations with the focus of aligning IT with business. The guideline was developed on behalf of the British government by the Office of Governance Commerce (OGC) in Norwich, England. ITIL V3 is the current version, which refreshed the collection and is the de facto global standard in the area of Service Management. It contains publicly available technical documentation for the planning, provision, and support of IT services.

IT Infrastructure Library (ITIL) is available at:

http://www.itil.org

The ITIL V3 publications are aligned with the new ITIL service life cycle. Part of the core publication structure is the introduction to the ITIL Service Life Cycle and the following five books (Figure 2-2 on page 48):

- *Service Strategy (SS)*
  provides guidance about how to design, develop, and implement IT services as strategic assets.
- *Service Design (SD)*
  provides guidance for the design and development of services and service management processes.

- *Service Transition (ST)*
  provides guidance for the improvement of capabilities and transitioning services into operations.
- *Service Operation (SO)*
  provides guidance about service delivery and support in order to ensure value for the customer and service provider.
- *Continual Service Improvement (CSI)*
  provides guidance for creating value for customers through better service management.

In the ITIL V3 publication, the Service management processes are applied across the new ITIL Service Life Cycle.

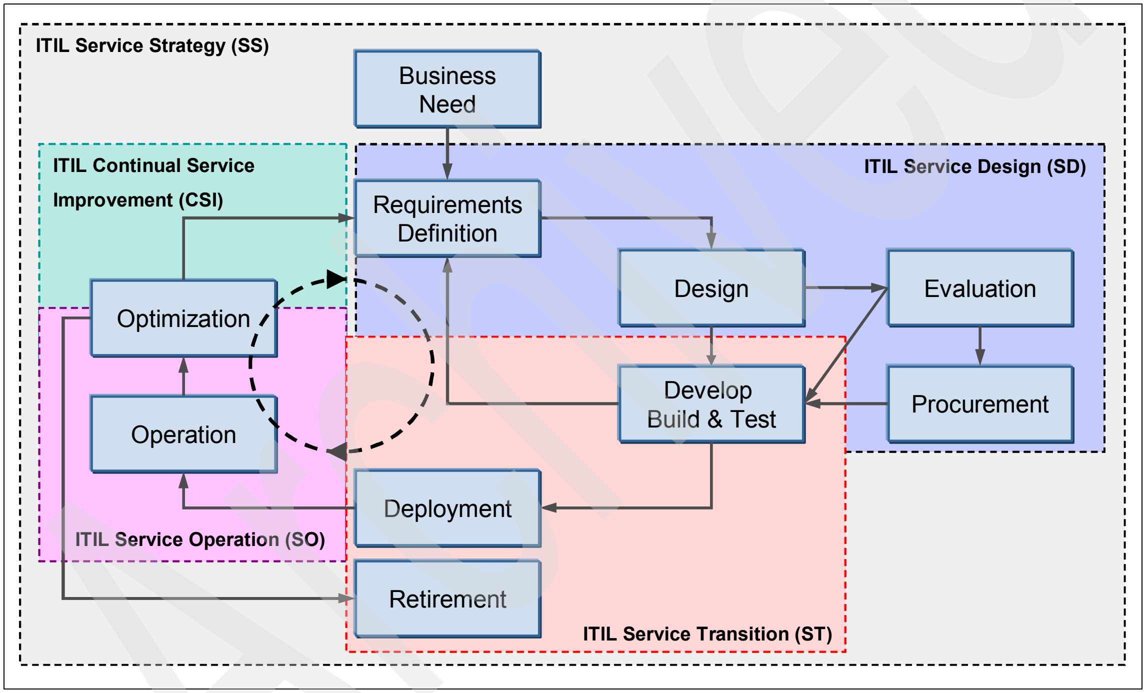


Figure 2-2 Service Management in the ITIL Service Life Cycle

ISO

The International Organization for Standardization (ISO) is the world's largest developer of standards. A Central Secretary in Geneva, Switzerland, coordinates a network of the national standards institutes of 155 countries, on the basis of one member per country.

The International Organization for Standardization (ISO) is available at:

http://www.iso.org

The ISO/IEC 20000-1:2005 Specification was developed for Information technology - Service management. It is divided in two parts:

- Part 1: Specification

  Defines the requirements for a service provider to deliver managed services. It is based on BS 15000-2, which has been superseded by ISO/IEC 20000.

  The adoption of an integrated process approach to effectively deliver managed services is emphasized in this specification. The target is to meet the business and customer requirements.

  Implementation and integration of service management processes provide the ongoing control, higher productivity, and opportunity for constant improvement. The usage of effective service management delivers high levels of customer service and customer satisfaction. It also recognizes that services and service management are essential to helping organizations generate revenue and be cost-effective.

- Part 2: Code of practice

  Represents an industry consensus on guidance to auditors and offers assistance to service providers planning service improvements or an audit against ISO/IEC 20000-1. ISO/IEC 20000-2:2005 is based on BS 15000-2, which has been superseded.

  The ISO/IEC 20000 series draws a distinction between organizational names and structures and best practices of processes, which are independent of organizational form or size. Both large and small service providers are supported.

  The service management processes focus on delivering the best possible service to meet a customer's business needs within agreed to resource levels. For example, the service management process strives for service that is professional, cost-effective, and with risks, which are understood and managed.

COBIT

Focused on helping organizations meet today's business challenges, the IT Governance Institute (ITGI) has published COBIT V4.1 - Control Objectives for Information and related Technologies (COBIT).

COBIT V4.1 is available at:

http://www.isaca.org

COBIT is an IT governance framework and supporting toolset with the focus on allowing managers to bridge the gap between control requirements, technical issues, and business risks. COBIT wants to enable clear policy development and good practices for IT control throughout organizations. COBIT plans to emphasize regulatory compliance, help organizations to increase the value attained from IT, enable alignment, and simplify implementation of the COBIT framework.

COBIT's IT control objectives are organized by the IT process; therefore, the framework provides a clear link among IT governance requirements, IT processes, and IT controls. COBIT focuses on what is required to achieve adequate management and control of IT and is positioned at a high level.

eTOM

Enhanced Telecom Operations Map (eTOM) is a framework of business processes in IT and telecommunications industry companies. It will be published by TeleManagement Forum (tmforum).

eTOM Release 7 is available at:

http://www.tmforum.org

The basic approach of eTOM involves telecommunications and IT companies often having to change data in a process chain to offer a service to a user. To get the highest possible quality for this service, it is necessary that all partners have the same view to the promised attributes of this service.

One requirement to reach this point is the transparency of the business processes for all involved companies. The easiest way to reach this target is to align the processes with standards. eTOM tries to build these standards.

The difference between ITIL and eTOM is the focus. ITIL focuses on professional operation and eTOM's focus is the provision of transparency services across different companies. It is possible to use ITIL and eTOM in combination. In addition to ITIL, a datamodel for each process is included with eTOM called Shared Information & Data Model (SID).

A tmforum concept called New Generation Operations Systems and Software (NGOSS) builds guidelines and specifications for SOA.

CMMI

Capability Maturity Model Integration (CMMI) is a process improvement approach that provides organizations with the essential elements of effective processes.

CMMI is available at:

http://www.sei.cmu.edu/cmmi/

CMMI can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes.

eSCM-SP

eSourcing Capability Model for Service Providers (eSCM-SP) V2 was developed by the IT Services Qualification Center (ITsqc) with a focus improving the sourcing relationships in the Internet-enabled economy.

eSCM-SP V2 is available at:

http://itsqc.cs.cmu.edu

Organizations are increasingly delegating their IT business activities to external service providers. In these business relationships, the management and meeting of client expectations is a major challenge.

There were three main purposes for developing eSCM-SP V2:

- Qualification of sourcing service providers
- Improvement of their ability to provide high quality sourcing services
- The possibility to differentiate themselves from the competition

IBM PRM-IT

One of the key concepts behind the IBM Process Reference Model for IT (PRM-IT V2) is that IT can be viewed as an essential component of any business, and that it can be managed as an asset.

The PRM-IT is an integrated collection of the processes involved in using Information Technology (IT) to assist businesses in carrying out many or all of their fundamental purposes. It describes - at a generic level - the activities that are performed in order for IT to provide value to the stakeholding business or businesses.

PRM-IT is used by Tivoli Services and IBM Global Services. IBM Tivoli Unified Process (ITUP) Composer V2.1.1 provides a Rational Method Composer plug-in for ITUP content used to implement PRM-IT. ITUP includes processes that are used within the scope of identifying nonfunctional requirements. The key processes are Service Level Management, Solution Requirements, and Solution Analysis and Design.

3

IBM SOA Foundation and the SOA Metadata Registry and Repository realization

This section introduces IBM service-oriented architecture (SOA) solution architecture - the IBM SOA Foundation. It describes the reference architecture, life cycle (model, assemble, deploy, and manage), and scenarios associated with the IBM SOA Foundation.

This provides the basis for the second part of this chapter, which describes metadata in an SOA environment. Metadata information domains are defined and used to illustrate SOA metadata use cases through the four states of the IBM SOA Foundation Life Cycle.

The third and final part of the chapter introduces an additional life cycle - the service life cycle. We use this life cycle to identify three stages: Service Development, Service Deployment, and Service Management. We then map IBM products to these stages.

3.1 IBM SOA Foundation

This section defines the IBM SOA Foundation, introduces the SOA Foundation Reference Architecture, describes the four life cycle stages of the SOA Foundation, and introduces the SOA Foundation scenarios.

3.1.1 The IBM SOA Foundation

The IBM SOA Foundation is an integrated, open standards-based set of IBM software, best practices, and patterns designed to provide what you need to get started with service-oriented architecture from an architectural perspective. The key elements of the IBM SOA Foundation are:

- The IBM SOA Foundation Reference Architecture
- The IBM SOA Foundation Life Cycle (model, assemble, deploy, and manage)
- The IBM SOA Foundation scenarios

The SOA Foundation scenarios (or simply SOA scenarios) are representative of common scenarios that use IBM products and solutions for SOA engagements. The SOA scenarios communicate the business value, architecture, and IBM open standards-based software used in the SOA scenario.

Note: For a more detailed explanation of the SOA Foundation, refer to *IBM SOA Foundation: An Architectural Introduction and Overview V1.0*, found at:

`http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf`

3.1.2 IBM SOA Foundation Reference Architecture

This section describes the SOA Foundation Reference Architecture, which includes the components and middleware services used by applications in the runtime environment.

Figure 3-1 on page 55 depicts the SOA Foundation Reference Architecture solution view used to decompose an SOA design.

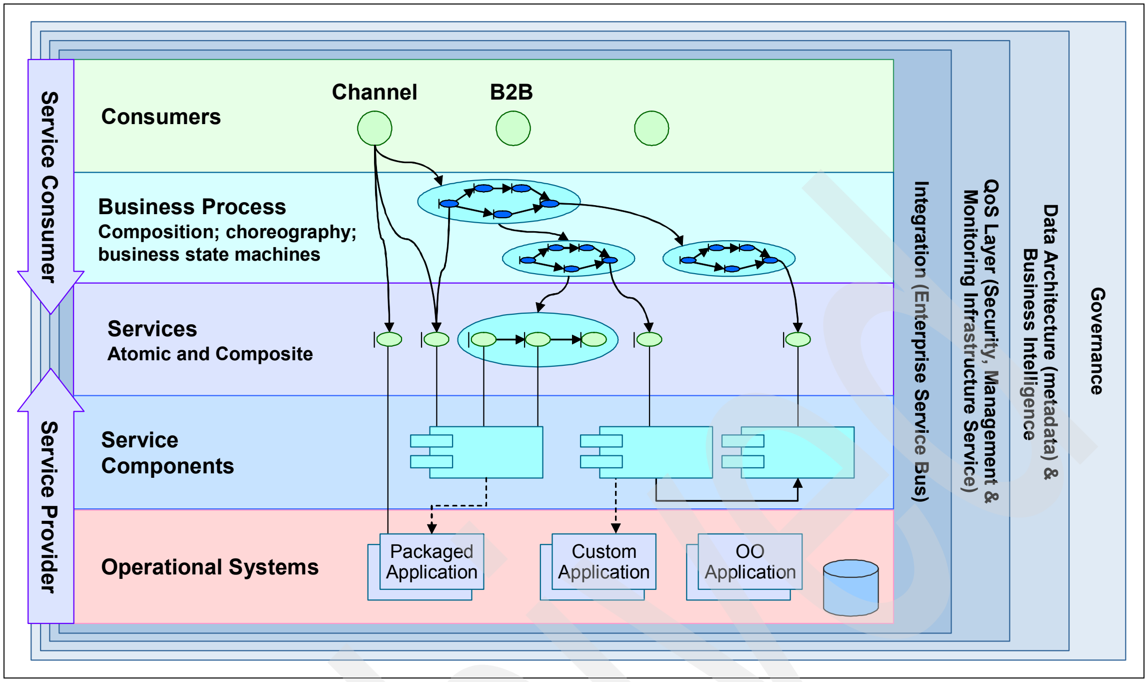


Figure 3-1 SOA Foundation Reference Architecture: Solution view

SOA puts a premium on the role of the *Enterprise Architect*, who is responsible for spanning between the business design and the information system that codifies that design.

When taking a top-down approach, the enterprise architect starts by identifying the business processes and business services used by business users. *Business users* are consumers of the processes and services. Business processes should be treated as compositions of other business processes and services and, thus, should be decomposed into their subordinate sub-processes and services.

> **Note:** This flow describes a top-down approach. Other variations include a bottom-up approach and the more common meet-in-the-middle approach.

Services and business processes are then detailed into service components. *Service components* include a detailed set of definition metadata used to describe the service to the information system. Services can be aggregated into module assemblies. The *module assemblies* are used to establish related design concerns and begin the planning to determine what teams will collaborate to implement the related services to be deployed as a single unit.

The resulting set of business process definitions, services, and schema will make up the logical architecture of the application. The enterprise architect then maps that logical architecture to a physical architecture. Figure 3-2 illustrates the SOA Foundation Reference Architecture: Middleware Services view.

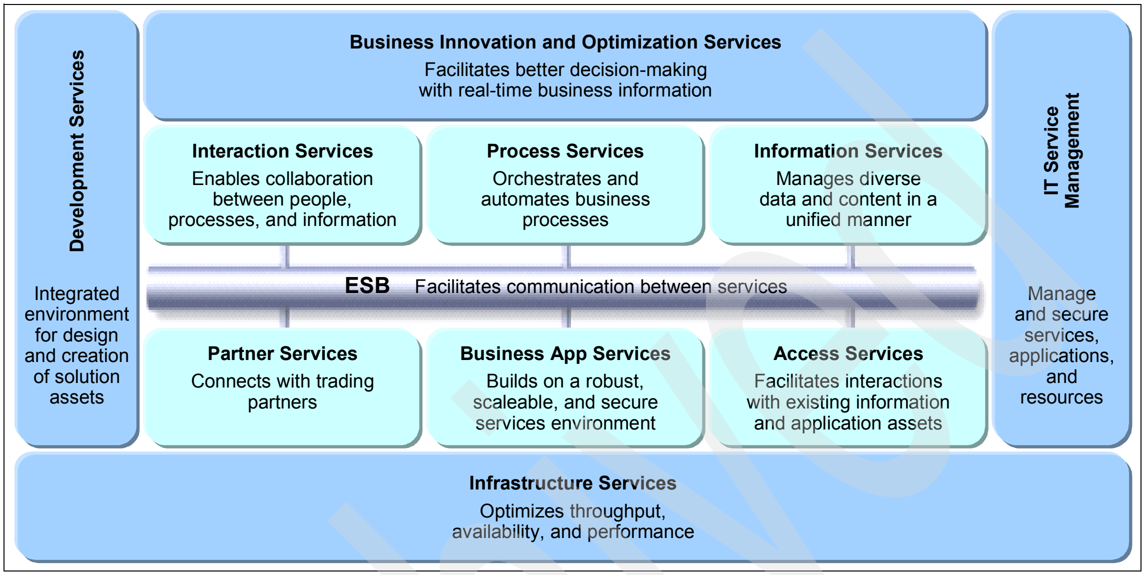


Figure 3-2 SOA Foundation Reference Architecture: Middleware Services view

We have included a summary description for each of the services found in the logical architecture displayed in Figure 3-2. The services found in the center of Figure 3-2 (Interaction, Process, Information, Partner, Business Application, and Access) are the core set of services used by applications within the runtime environment when deployed. The other services (outer services) displayed in Figure 3-2 are used in support of the core services.

Core components of the logical architecture

This section includes a brief description of the following core components of the logical architecture:

- Interaction services
- Process services
- Business application services
- Information services
- Access services
- Partner services

Interaction services

Interaction services provide the capabilities that are required to deliver IT functions and data to users, meeting their specific preferences.

Process services

Process services provide the required control capabilities for managing the flow and interactions of multiple services in ways that implement business processes.

Business application services

Business application services are called by service consumers. Service consumers include other components in the logical architecture, such as a portal or business processes.

Information services

Information services provide the capabilities necessary to federate, replicate, and transform disparate data sources.

Access services

Access services provide bridging capabilities among core applications, prepackaged applications, enterprise data stores, and the Enterprise Service Bus (ESB) to incorporate services that are delivered through existing applications into an SOA.

Partner services

Partner services provide the document, protocol, and partner management capabilities for business processes that involve interactions with outside partners and suppliers.

Supporting components of the logical architecture

This section includes a brief description of the supporting components of the SOA Foundation logical architecture used in support of the core components:

- Enterprise Service Bus
- Business innovation and optimization services
- Development services
- IT service management
- Infrastructure services

Enterprise Service Bus

The Enterprise Service Bus (ESB), or simply *bus,* provides an infrastructure that removes the direct connection dependency between service consumers and providers. Consumers connect to the bus and not the provider that actually implements the service. This type of connection further decouples the consumer from the provider. A bus also implements further value-add capabilities, such as security and delivery assurance. We recommend that you implement these capabilities centrally within the bus at an infrastructure level rather than within the application. The primary driver for an ESB, however, is that it increases decoupling between service consumers and providers.

Business innovation and optimization services

Business innovation and optimization services are primarily used to represent the tools and the metadata structures for encoding the business design, including the business policies and objectives.

Business innovation and optimization services exist in the architecture to help capture, encode, analyze, and iteratively refine the business design. The services also include tools to help simulate the business design. The results are used to predict the effect of the design, including the changes that the design will have on the business.

Development services

Development services encompass the entire suite of tools and aids that are needed to construct an SOA-based application:

- Architecture tools
- Development tools
- Visual composition tools
- Assembly tools
- Methodologies
- Debugging aids
- Instrumentation tools
- Asset repositories
- Discovery agents
- Publishing mechanisms

IT service management

After the application has been deployed to the runtime environment, it has to be managed along with the IT infrastructure on which it is hosted. IT service management represents the set of management tools that are used to monitor your service flows, the health of the underlying system, the utilization of resources, the identification of outages and bottlenecks, the attainment of service goals, the enforcement of administrative policies, and recovery from failures. IT service management is a part of SOA management.

Infrastructure services

Infrastructure services form the core of the information technology runtime environment used for hosting SOA applications. These services provide the ability to optimize throughput, availability, performance, and management.

3.1.3 IBM SOA Foundation Life Cycle

IBM clients have indicated that they think of SOA in terms of a life cycle. The IBM SOA Foundation includes the following life cycle phases (Figure 3-3):

- Model
- Assemble
- Deploy
- Manage

There are a couple of key points to consider about the SOA life cycle:

- The SOA life cycle phases apply to all SOA projects.
- The activities in any part of the SOA life cycle can vary in scale and the level of tooling used depending on the stage of adoption.

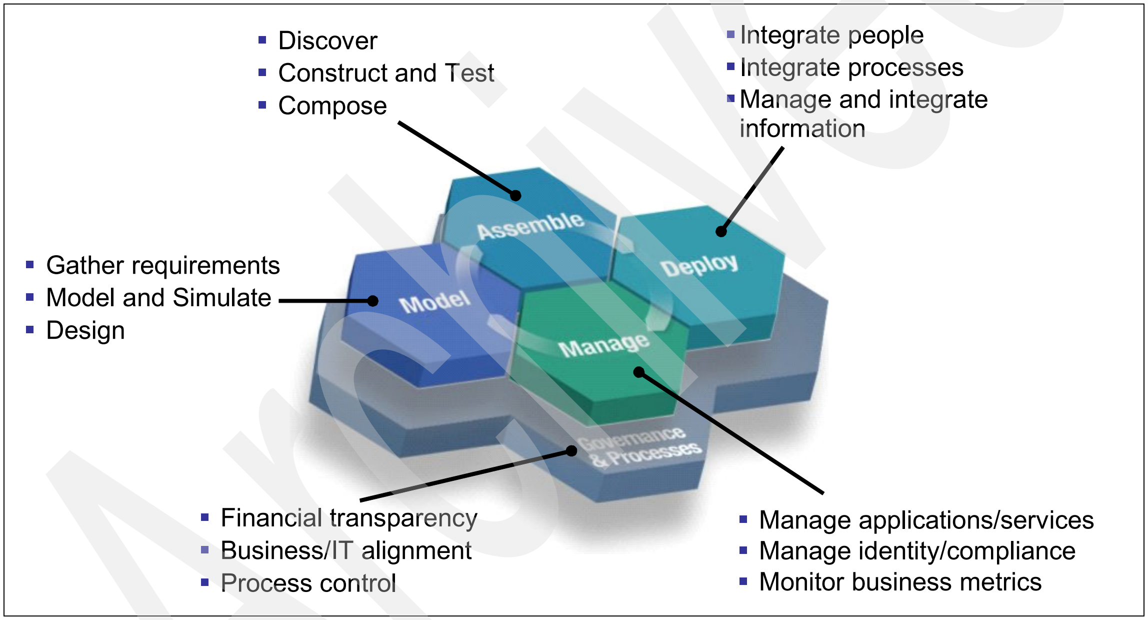


Figure 3-3 IBM SOA Foundation Life Cycle

Model

Modeling is the process of capturing the business design from an understanding of business requirements and objectives. Business requirements are translated into a specification of business processes, goals, and assumptions for creating a model of the business. Many businesses do not go through a formal modeling exercise. In some cases, businesses that do perform modeling use primitive techniques, such as drawing the design in Microsoft® Visio® or using text documents.

Capturing the business design using a sophisticated approach that includes the use of specialized tooling enables you to perform what-if scenarios with various parameters that the business might experience. The process can then be simulated using those parameters to predict the effect that the process will have on the business and IT systems. If the achieved results do not match the business objectives, the process definition can be refined.

The model also captures key performance indicators, such as business metrics that are important measurements of your business. For example, key performance indicators can include a measure of the new accounts that you have opened in a given month. These key performance indicators are input to the assembly of the application. In addition, the indicators can be monitored in production to capture the critical data to measure whether the objectives are being met.

Assemble

The business design is used to communicate the business objectives to the IT organization that will assemble the information system artifacts that implement the design. The enterprise architect works closely with the business analyst to convert the business design into a set of business process definitions, as well as activities used to derive the required services from the activity definitions. The enterprise architect and business analyst work with the software architect to further develop the design of the services.

During the process of resolving the design and implementation of the modeled business processes and services, a search of existing artifacts and applications needs to be performed in an effort to find components that meet the needs of the design. Some applications will fit perfectly, some will have to be re-factored, and some will have to be augmented to meet the requirements of the design.

These existing assets must be rendered as services for assembly into composite applications. Any new services that are required by the business design have to be created. Software developers need to use the SOA programming model to create these new services.

Finally, the assemble phase includes applying a set of policies and conditions to control how your applications operate in the production runtime environment. For example, these policies and conditions include business and government regulations. In addition, the assemble phase includes critical operational characteristics, such as packaging deployment artifacts, localization constraints, resource dependency, integrity control, and access protection.

Deploy

The deploy phase of the life cycle includes a combination of creating the hosting environment for the applications and the deployment tasks of those applications.

This includes resolving the application's resource dependencies, operational conditions, capacity requirements, and integrity and access constraints.

A number of concerns are relevant to the construction of the hosting environment, including the presence of the existing hosting infrastructure supporting applications and preexisting services. Beyond that, you need to consider appropriate platform offerings for hosting the user interaction logic, business process flows, business services, access services, and information logic.

Manage

The manage phase includes the tasks, technology, and software used to manage and monitor the application assets, such as services and business processes that are deployed to the production runtime environment.

Monitoring is a critical element of ensuring that the underlying IT systems and applications are up and running to maintain the service availability requirements of the business. Monitoring also includes monitoring performance of service requests and timeliness of service responses. In addition, monitoring includes maintaining problem logs to detect failures in various services and system components, as well as localizing failures and restoring the operational state of the system.

Managing the system also involves performing routine maintenance; administering and securing applications, resources, and users; and predicting future capacity growth to ensure that resources are available when the demands of the business call for them. The security domain includes topics, such as authentication, single sign-on, authorization, federated identity management, and user provisioning.

The manage phase also includes managing the business model, tuning the operational environment to meet the business objectives expressed in the business design, and measuring success or failure to meet those objectives. SOA is distinguished from other styles of enterprise architecture by its correlation between the business design and the software that implements that design, and its use of policy to express the operational requirements of the business services and processes that codify the business design. The manage phase of the life cycle is directly responsible for ensuring that those policies are being enforced and for relating issues with that enforcement back to the business design.

3.1.4 IBM SOA Foundation scenarios

The SOA Foundation scenarios (or simply SOA scenarios) are representative of common scenarios of using IBM products and solutions for SOA engagements.

The SOA scenarios quickly communicate the business value, architecture, and IBM open standards-based software used in the SOA scenario. The SOA scenarios can be implemented as part of an incremental adoption of SOA growing from one scenario to using elements of multiple scenarios together. *Realizations* are used to provide more specific solution patterns and IBM product mappings within the SOA scenarios. (A realization is an example business case that describes a client situation and the solution.)

The SOA scenarios can be used as a reference architecture implementation (starting point) to accelerate the SOA architecture and implementation of your client scenario. Figure 3-4 illustrates the SOA scenarios and entry points.

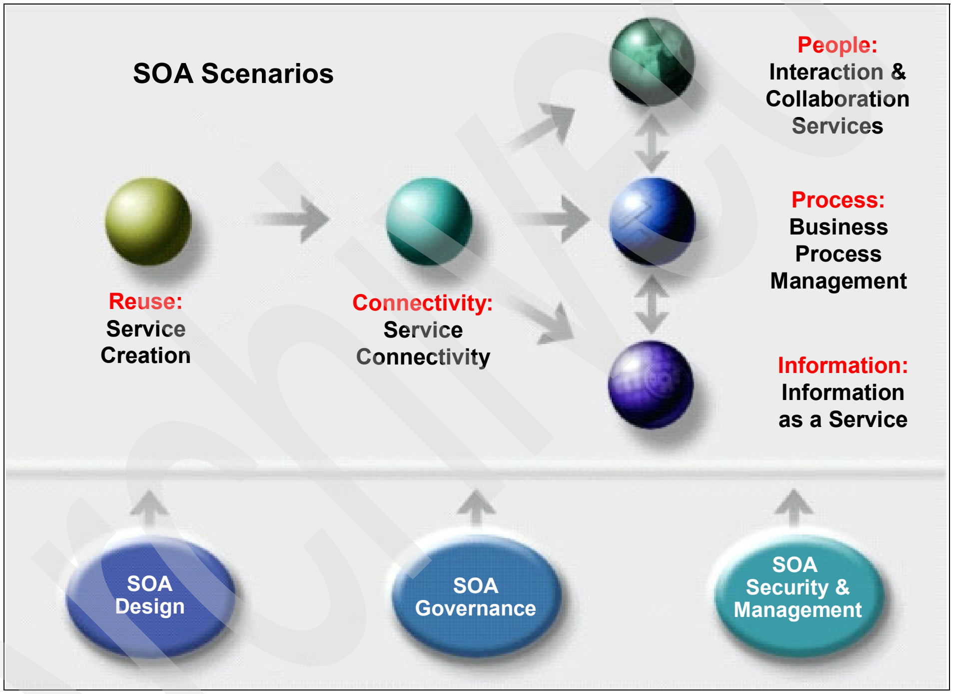


Figure 3-4 SOA scenarios and entry points

Figure 3-4 displays the SOA scenarios (Service Creation, Service Connectivity, Interaction and Collaboration Services, Business Process Management, and Information as a Service), and the relationships among the scenarios.

We have included examples of how the scenarios can be used together and adopted incrementally. For example, it is common that the other scenarios will include service creation and often want connectivity. In addition, the scenarios can be used together, such as a portal accessing a business process or a portal accessing an information service through an ESB from a service consumer.

SOA Design, SOA Governance, SOA Security, and SOA Management can be used in each of the SOA scenarios based on client requirements.

3.2 Metadata in an SOA environment

Metadata in an SOA environment is a type of data that describes an SOA environment or, in shorter terms, data about data of an SOA environment.

This section describes:

- Metadata overview
- Metadata information domains
- SOA repository roles
- Metadata in the model, assemble, deploy, and manage stages

3.2.1 Metadata overview

As the market becomes increasingly aware of the benefit that metadata can provide in the enterprise, clients are demanding that vendors take a responsible approach to metadata management. Metadata enables key functionality across IT - and the enterprise as a whole - by providing better functionality.

Making the benefits a reality becomes a challenge both at the individual tool level and at the wider IT landscape scope. Individual tools must produce good metadata; IT must be able to manage that metadata across the entire scope of the enterprise.

To specify how metadata is used in an SOA, it is first necessary to define which kinds of metadata are in scope and which are out of scope (Figure 3-5 on page 64).

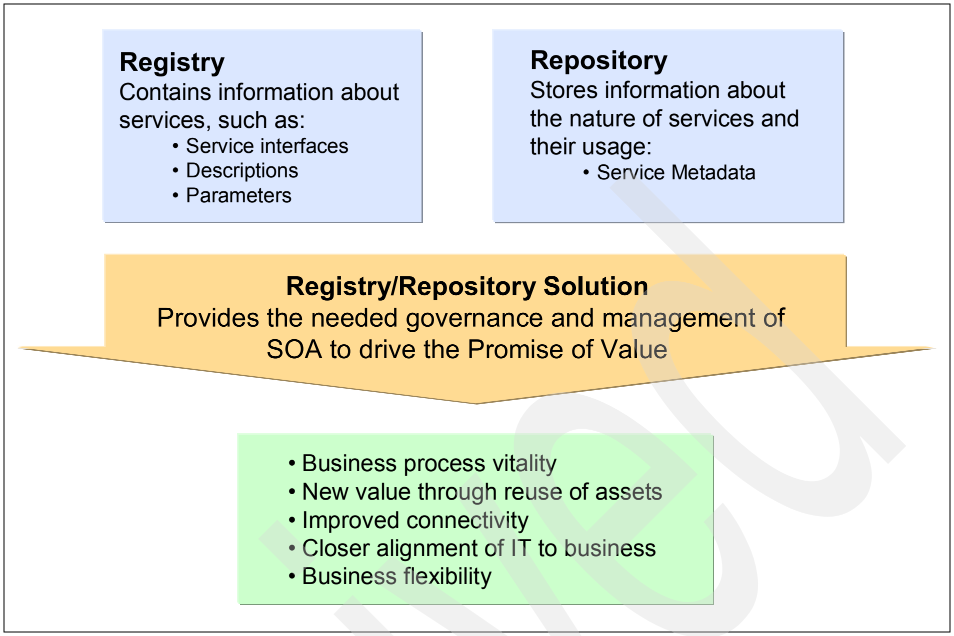


Figure 3-5 Metadata store location and impact

Metadata in scope of SOA

Metadata in scope of SOA is information about services, service-oriented applications, and the context of these within SOA. Service metadata is incrementally produced and transformed as services and service-oriented applications advance through the SOA life cycle.

Examples:

- Service, process, data models, and designs
- Topology, deployment models, and designs
- Service configurations
- Service Level Agreements (SLAs) and their results

Metadata in scope of SOA includes:

- The metadata artifacts related to services in any life cycle stage and in any problem domain
- The data that establishes context of those artifacts from an overall SOA perspective

3.2.2 Metadata information domains

IBM software solutions that participate in SOA use a number of information domains (shown in Figure 3-6). These domains serve different purposes and support different perspectives and usage contexts within the overall service life cycle.

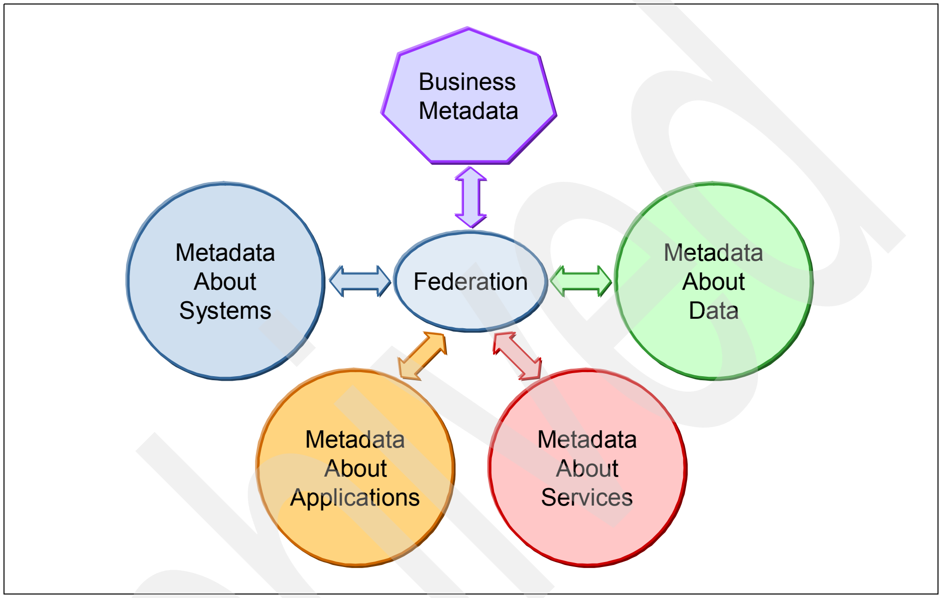


Figure 3-6 Metadata information domains

Each of these domains meets the specific needs of its respective user requirements. If user requirements need additional information, the information domain can integrate with other information domains to bring related metadata into the domain context. This approach of federating information among the information domains is core to the IBM approach.

The metadata information domains shown in Figure 3-6 are:

- Business Metadata

  The Business Metadata domain is used in the Model and Manage SOA life cycle stages.

  Typically stored data includes: Business models, taxonomies, ontologies, and glossaries:

  - Associated IBM products:
    - IBM Industry Models

  - IBM WebSphere Business Glossary
  - IBM Rational Asset Manager

- Metadata about Data

  The Metadata about Data domain is used in the Model, Assemble, Deploy, and Manage SOA life cycle stages.

  Typically stored data includes: Data models, data warehouse models, data definitions, database definitions, XML Schema Definition (XSD), and master data definitions:

  – Associated IBM products:

    - IBM Information Server
    - IBM WebSphere Customer Center
    - IBM WebSphere Product Center
    - IBM Rational Data Architect
    - IBM Entity Analytic Solutions
    - IBM WebSphere Federation Server

- Metadata about Services

  The Metadata about Services domain is used in the Model, Assemble, Deploy, and Manage SOA life cycle stages.

  Typically stored data includes: Service descriptions, Web Services Description Language (WSDL), XML Schema Definition (XSD), Service Component Description Language (SCDL), service policies, service integration configurations, service performance, and service health:

  – Associated IBM products:

    - IBM WebSphere Services Registry and Repository
    - IBM WebSphere Enterprise Service Bus
    - IBM WebSphere Business Modeler
    - IBM Tivoli Composite Application Manager for SOA
    - IBM WebSphere Message Broker

- Metadata about Applications

  The Metadata about Applications domain is used in the Model, Assemble, and Deploy SOA life cycle stages.

  Typically stored data includes: Requirements, design models, assets, source code, binaries, process definitions, and application configurations:

- Associated IBM products:
  - IBM Rational ClearCase®
  - IBM Rational ClearQuest®
  - IBM Rational Asset Manager
  - IBM WebSphere Asset Analyzer
  - IBM CICS Interdependency Analyzer
  - IBM Asset Transformation Workbench
  - IBM Rational Application Developer

- Metadata about Systems

  The Metadata about Systems domain is used in the Model, Deploy, and Manage SOA life cycle stages.

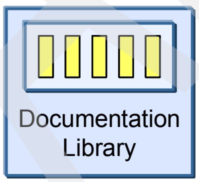
  Typically stored data includes: topology models and realizations, deployment models and realizations, deployed application and service configurations, IT assets, configuration items, observed services, process structures, and roles:

  - Associated IBM products:
    - IBM Rational Build Forge®
    - IBM Tivoli Configuration Manager
    - IBM Tivoli Application Discovery Dependency Manager
    - IBM IT Service Manager

3.2.3 SOA repository roles

This section explains the roles that registries and repositories play throughout the SOA life cycle.

Documentation Library



A *Documentation Library* stores and manages documentation artifacts (specifications, forms, descriptions, and so forth) that are associated with a service. These libraries often belong to governance processes.

Typically, data is stored during the Model stage and retrieved during the Manage stage of the SOA life cycle.

The current IBM software products that are normally used as a Document Library repository in an SOA life cycle are:

- IBM Rational ClearCase
- IBM Rational Asset Manager
- IBM WebSphere Service Registry and Repository

Asset Management

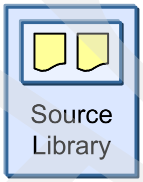
Asset Management stores, manages, and governs assets (including services).

Typically, data is stored during the Model, Assemble, Deploy, and Manage stages and retrieved during the Model, Assemble, and Manage stages of the SOA life cycle.

The current IBM software product that is often used as an Asset Management repository in an SOA life cycle is:

- IBM Rational Asset Manager

Source Library



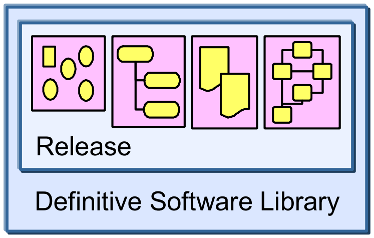
A *Source Library* stores, manages, and governs development artifacts.

Typically, data is stored during the Model and Assemble stages and retrieved during the Model and Assemble stages of the SOA life cycle.

The current IBM software product that is normally used as a Source Library repository in an SOA life cycle is:

- IBM Rational ClearCase

Definitive Software Library



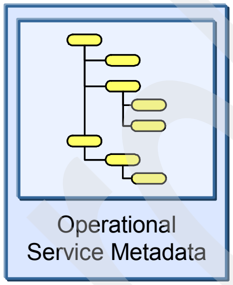
A *Definitive Software Library* stores, manages, and governs production software artifacts and configurations.

Typically, data is stored during the Model, Assemble, Deploy, and Manage stages and retrieved during the Deploy stage of the SOA life cycle.

The current IBM software product that is normally used as a Definitive Software Library repository in an SOA life cycle is:

- IBM Rational ClearCase

Operational Service Metadata



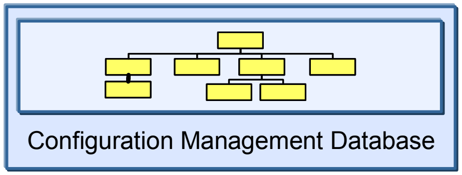
Operational Service Metadata stores, manages, and governs metadata pertaining to services.

Typically, data is stored during the Assemble, Deploy, and Manage stages and retrieved during the Model, Assemble, and Manage stages of the SOA life cycle.

The current IBM software product that is normally used as an Operational Service Metadata repository in an SOA life cycle is:

- IBM WebSphere Service Registry and Repository

Configuration Management Database



A *Configuration Management Database* stores information about systems.

Typically, data is stored and retrieved during the Deploy and Manage stages of the SOA life cycle.

The current IBM software product that is normally used as a Configuration Management Database repository in an SOA life cycle is:

- IBM Tivoli Change and Configuration Management Database

Service Observations

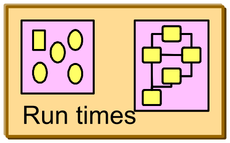
Service Observation stores and manages operational information collected through operational monitoring.

Typically, data is stored during the Manage stage and retrieved during the Model, Assemble, and Manage stages of the SOA life cycle.

The current IBM software products that are normally used as a Operational Service Metadata repository in an SOA life cycle are:

- IBM Tivoli Composite Application Managers (family)
- IBM Tivoli Monitoring (family) including Tivoli Enterprise Management Server

Run times



A *Run time* consumes and allows administrative changes to the operational metadata. A Run time is not strictly considered to be a metadata store.

Typically, data is stored during the Deploy and Manage stages and retrieved during the Manage stage of the SOA life cycle.

The current IBM software products that are normally used as Run time repositories in an SOA life cycle are:

- IBM WebSphere Application Server (family)
- IBM WebSphere Process Server (family)
- IBM WebSphere Business Services Fabric
- IBM WebSphere MQ Workflow for Multiplatforms
- IBM WebSphere Message Broker
- IBM WebSphere Enterprise Service Bus
- IBM DataPower® Appliances
- IBM CICS Transaction Server

3.2.4 Metadata in the Model stage

After identification of the business domain, the assignment of ownership and the the allocation of funding business services, business processes, and relevant data have to be identified and modeled.

The driver in this stage is the business scope with its business goals, business requirements, and business use cases. Based on this driver, there are normally business and design models implemented.

Two use cases, *Identify Service* and *Model Service* (Figure 3-7 on page 72), are described in this section. Based on the adoption level of the SOA life cycle, the following interaction tasks are covered by these use cases:

- Find/retrieve/use/create assets
- Find/retrieve/use/create models
- Find/retrieve/use/create service descriptions

- Record use of artifacts
- Store new models
- Store new model assets
- Analyze impact of change
- Specify business process model
- Submit business process model
- Submit service specification
- Model deployment topology
- Find/retrieve/create requirements
- Store service specification

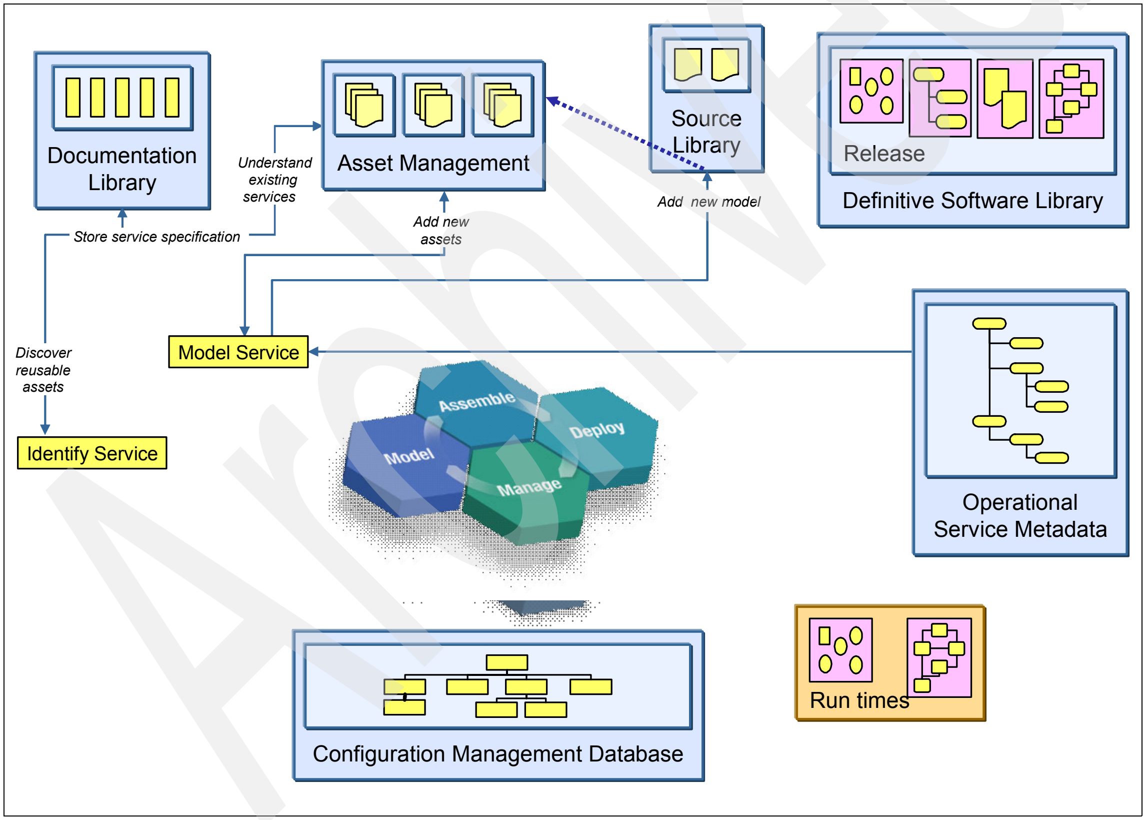


Figure 3-7 Metadata in the Model stage of the SOA life cycle

Use case: Identify Service

In the use case *Identify Service,* an Asset Management store is searched for reusable assets. After understanding and discovering a reusable asset, this metadata is stored as a service specification in a Documentation Library.

if there is no reusable asset, the next step in the model stage will be the use case *Model Service*. If there is a reusable asset, the next step is the assemble stage.

Use case: Model Service

Metadata from an Operational Service Metadata store is used to find services that the service being modeled will consume. Based on the metadata out of the Operational Service Metadata store, a new service is created in the use case *Model Service*.

The new metadata of this service is added to a Source Library. This metadata is associated with the new assets, which are added to an Asset Management store.

3.2.5 Metadata in the Assemble stage

In the Assemble stage of the SOA life cycle, the following metadata has to be created and consumed:

- Metadata about the architecture design
- Metadata about the service development
- Metadata about the functionality tests

The use case, *Develop Service* (Figure 3-8 on page 74), is described in this section. The following interaction tasks are covered by this use case:

- Find/retrieve/use artifacts
- Record use of artifacts
- Store new artifacts
- Classify/manage/govern metadata
- Add new files to a release
- Submit release
- Submit service tests

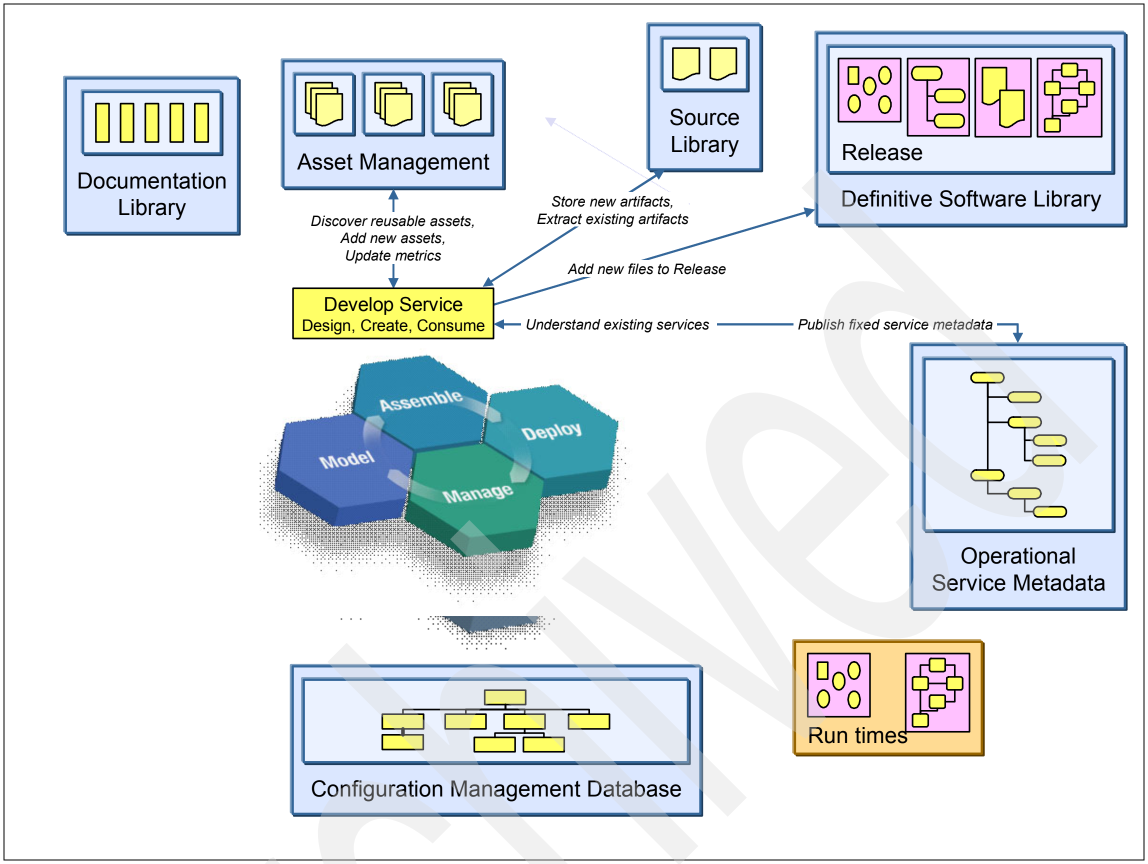


Figure 3-8 Metadata in the Assemble stage of the SOA life cycle

Use case: Develop Service

After the business service is identified or modeled in the model stage of the SOA life cycle, it has to be developed in this use case.

By understanding existing services out of an Operational Service Metadata store and discovering reusable assets in an Asset Management store, an architecture will be designed and added as a new asset in the Asset Management store.

The creation of the new service is based on the discovery of reusable assets out of an Asset Management store. Based on this metadata, existing artifacts have to be extracted out of a Source Library.

After this consuming of existing metadata, the development and test of the new service will bring new metadata for storing:

- New assets in an Asset Management store
- New artifacts in a Source Library
- New fixes to releases in a Definitive Software Library store
- Fixed service metadata in an Operational Service Metadata store

3.2.6 Metadata in the Deploy stage

During the Assemble stage, the new metadata was stored in different repositories. In the Deploy stage, the service has to be deployed and enabled.

Two use cases, *Deploy Service* and *Enable Service* (Figure 3-9 on page 76), are described in this section. Based on the adoption level of the SOA life cycle, the following interaction tasks are covered by these use cases:

- Retrieve contents of release
- Perform production build
- Package build results
- Distribute package
- Install package into run times
- Publish initial endpoint metadata
- Annotate, manage, and govern endpoint metadata
- Record changes made to the production environment
- Discover services
- Discover services metadata
- Publish intermediary endpoint metadata
- Associate service policies
- Update configuration in packaged applications
- Activate application/service

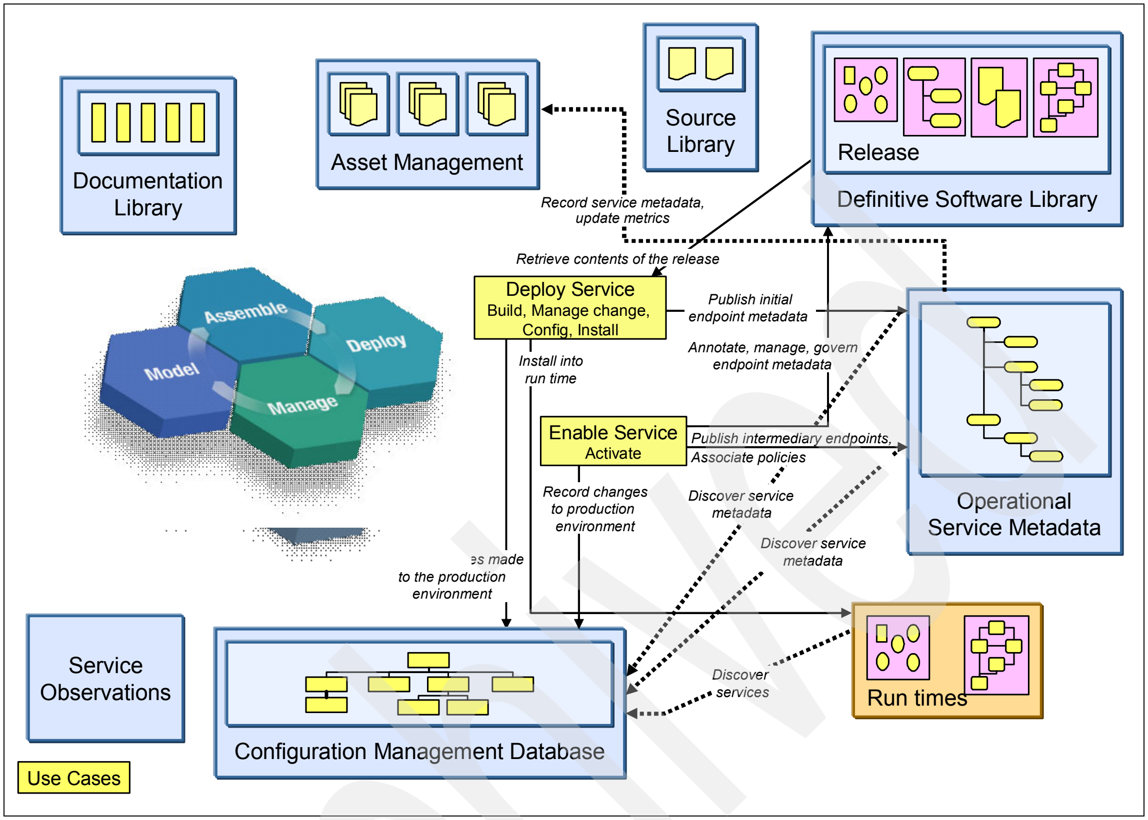


Figure 3-9 Metadata in Deploy stage of SOA life cycle

Use case: Deploy Service

In this use case, a new package is built, configuration and installation are performed, and all the changes are managed.

The new package is built, beginning with the retrieval of content about the developed release from a Definitive Software Library. The final package is then installed into a Run time.

The initial endpoint metadata is then published to an Operational Service Metadata store. In addition, service metadata is recorded in an Asset Management store and the metrics are updated there.

The last step in this use case is the recording of changes made to the production environment in a Configuration Management Database.

Use case: Enable Service

This step focuses on post-installation configuration and provisioning activities. When these tasks have been completed, the deployed service is activated.

By publishing the intermediary endpoints and associating the policies in a Operational Service Metadata store, the service is ready to use. Changes made to the production environment are recorded in a Configuration Management Database.

3.2.7 Metadata in the Manage stage

The knowledge of all relevant metadata and the understanding of the system build are fundamental for the Manage stage of the SOA life cycle.

The use case, *Detect Problem* (Figure 3-10 on page 78), is described in this section. Based on the adoption level of the SOA life cycle, the following interaction tasks are covered by this use case:

- Capture diagnostics
- Assess service performance
- Understand service relationships
- Understand system structure

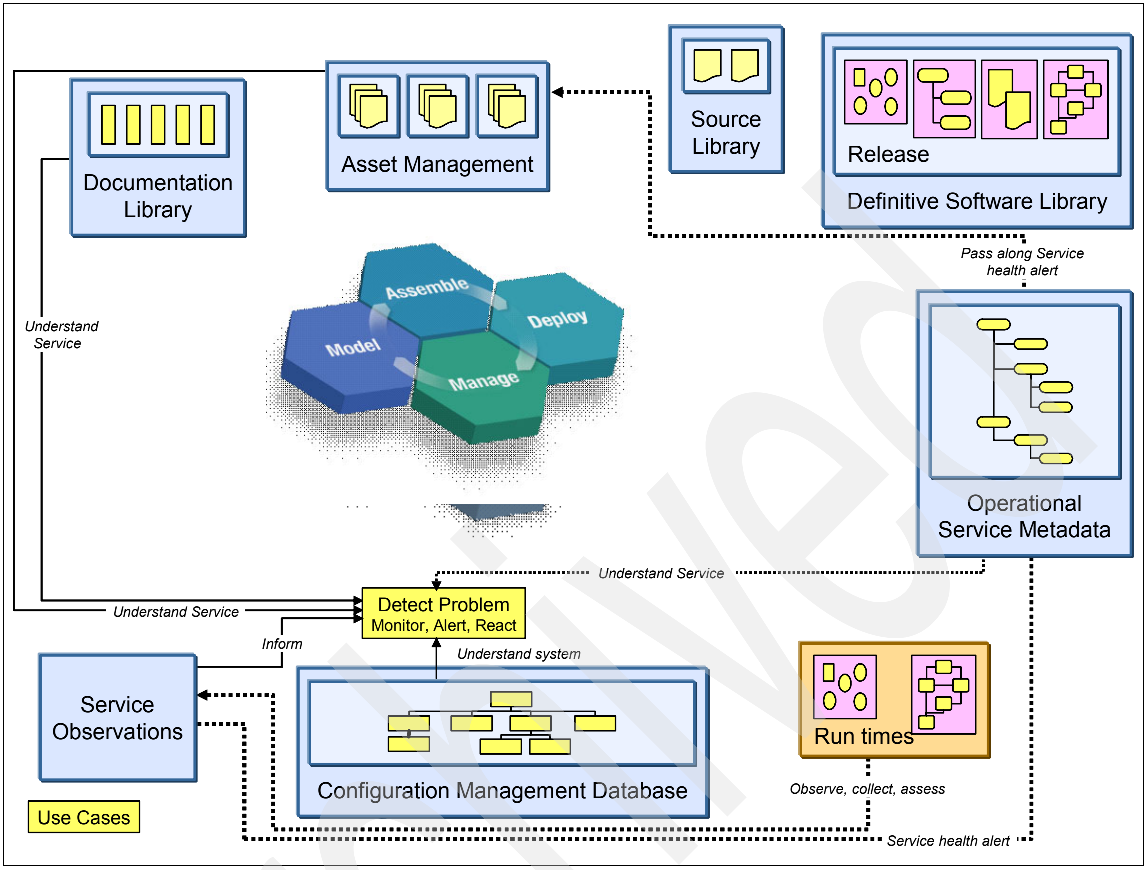


Figure 3-10 Metadata in the Manage stage of the SOA life cycle

Use case: Detect Problem

The use case, *Detect Problem*, is based upon the understanding of the services, the systems, and their coherences. The metadata from a Document Library, Asset Management store, Operational Service Metadata store, and Configuration Management Database build the knowledge base for this understanding.

The monitoring solution, which generates the metadata in a Service Observation store, observes the Run times, collects the data, and assesses the health status. In the case of a service performance situation, an Operational Service Metadata store and an Asset Management store are updated.

3.3 Service life cycle

This section introduces an additional life cycle called the *service life cycle*. The service life cycle consists of three primary stages: Service Development, Service Deployment, and Service Management. These stages build upon the Model, Assemble, Deploy, and Manage stages of the SOA life cycle.

Using the service life cycle, we can identify repositories and the interactions between them. These interactions need to be identified for an organization based on their SOA adoption level and the governance processes defined around these interactions.

This section explores the service life cycle, maps it to the SOA life cycles, and identifies some primary and supporting products for implementing the service life cycle.

The service life cycle is shown in Figure 3-11.

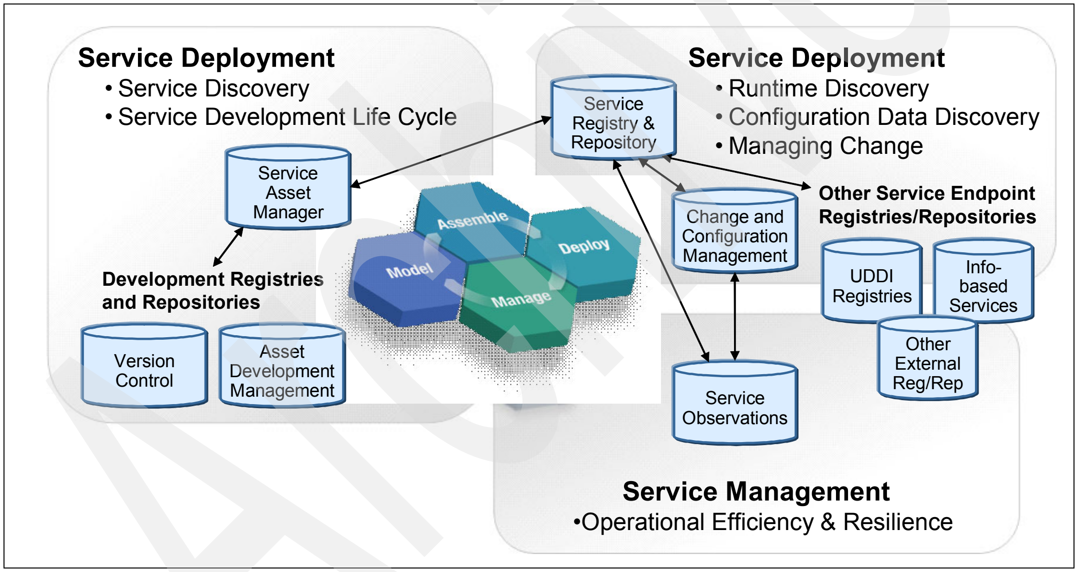


Figure 3-11 Mapping of SOA life cycle to service life cycle

The phases of the service life cycle and what they consist of are:

- Service Development

  This phase covers the Model and Assemble stages of the SOA life cycle and is associated with the Service Discovery and Service Development life cycle for a service. Types of roles for which repositories will be used during this phase with the associated governance processes include:

  – *Asset development management* for the storage of assets associated with the service, which can include process diagrams, requirements, and code
  – *Version control* for maintaining and managing differing versions and changes for the support of the asset management
  – *Service asset manager* to discover, develop, and reuse service assets

- Service Deployment

  This phase covers the Deploy stage of the SOA life cycle. The roles of the repositories include:

  – *Service registry and repository* for runtime discovery of services that are deployed in the environment
  – *Change and configuration management* for capturing and enforcing the authorized state of the services in the overall system
  – *Other service endpoint registries and repositories* that might be required for information. These can include UDDI registries, information-based registries, or external registries and repositories.

- Service Management

  This phase covers the Manage stage of the SOA life cycle associated with looking at the operational efficiency and resilience of the service. The roles of the repositories include:

  – *Service observations* for capturing information about the service based on IT-defined and business-defined triggers

Support of each of these repository roles can be covered across one or many repositories in an organization with the likelihood that repositories will cover multiple roles.

3.3.1 Mapping products to the service life cycle

Implementing the repositories in an organization can be performed in many ways depending on the level of SOA maturity, based on culture or budget and other people and process-related criteria. The only consideration that we will discuss is how these repositories map to the technology using IBM products to implement SOA governance.

Using the service life cycle approach and mapping to the SOA life cycle, the products identified for SOA governance are shown in Figure 3-12 on page 81.

Note: The selection of products described within this section is not definitive. They are the products that we used throughout this IBM Redbooks publication to illustrate SOA governance and management for our particular scenario.

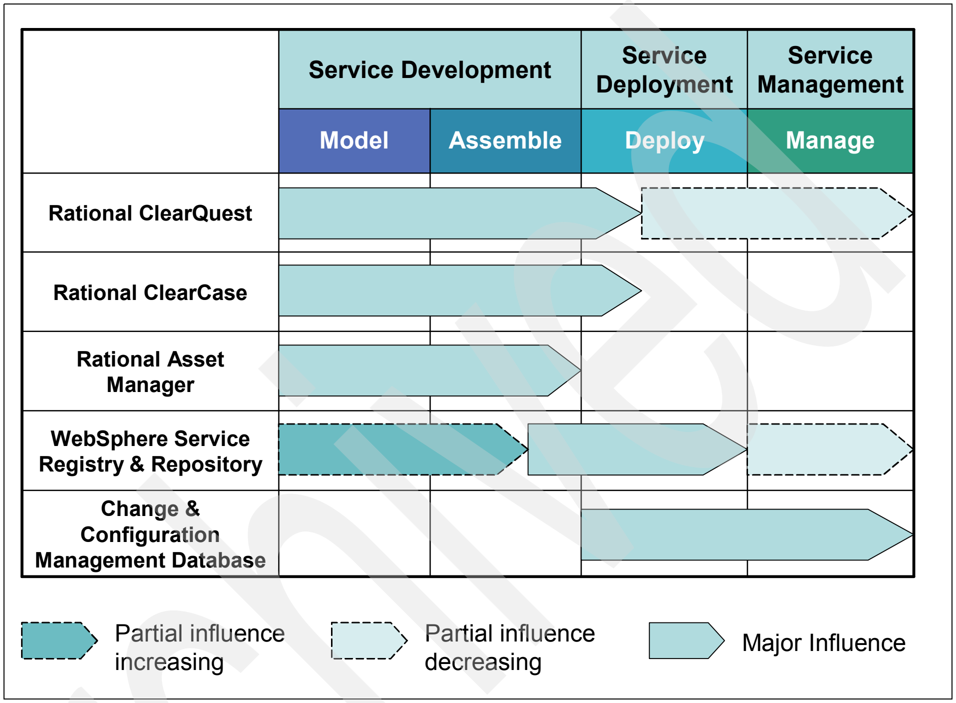


Figure 3-12 Mapping of products to the SOA and service life cycles

Note: Not all the products map one-to-one to the service life cycles but overlap partially or fully over some of the other cycles, indicating products that target more than one phase of the service life cycle.

We next show you how the products map to the repository roles defined in Table 3-1 on page 82.

Table 3-1 Products supporting each repository role across the service life cycle

Service life cycle	Repository role	Primary products	Supporting products
Service Development	Asset development management	► Rational Asset Manager	► Rational ClearCase ► Rational ClearQuest
	Version control	► Rational ClearCase ► Rational ClearQuest	N/A
	Service asset management	► Rational Asset Manager	► WebSphere Services Registry and Repository
Service Deployment	Service registry and repository	► WebSphere Services Registry and Repository	N/A
	Change and configuration management	► Tivoli Change and Configuration Management Database	► Tivoli Application Dependency Discovery Manager ► IT Service Management ► Tivoli Provisioning Manager ► Rational ClearCase ► Rational ClearQuest
Service Management	Service observation	► Tivoli Enterprise Management Server ► Tivoli Composite Application Manager for SOA	► Tivoli Change and Configuration Management Database ► WebSphere Services Registry and Repository

Note: The products identified against each repository role are not an exhaustive list but have been scoped for the purposes of this IBM Redbooks publication to ensure that coverage is provided for most roles. However, this does show how one product can perform multiple repository roles in establishing an SOA governance solution.

3.3.2 Product mapping clarifications

The products depicted in Figure 3-12 on page 81 as mapping to other phases of the SOA life cycle, require further explanation on the rationale behind each of them to understand what role they play in that phase. We provide those explanations now:

- Rational ClearQuest is shown to map fully to a part of the Service Deployment phase indicating the role that it plays in transitioning the product into a deployment environment and passing information to products used in this phase.

The partial role in the Service Deployment and Service Management stages indicates how Rational ClearQuest is used to identify defects, which are fed back to the Service Development phase.

- Rational ClearCase is shown to map fully to a part of the Service Deployment phase indicating the role that it plays in migrating assets into the deployment phase and in its role as a Definitive Software Library.
- WebSphere Service Registry and Repository maps partially to the Service Development phase indicating the role that it plays in the service discovery during the SOA life cycle Model and Assemble phase.

  The overlap in the SOA life cycle Assemble phase indicating a major influence identifies the relationship between Rational Asset Manager and WebSphere Service Registry and Repository. After an service asset is reviewed in Rational Asset Manager, a process of deploying it to WebSphere Service Registry and Repository can occur prior to full deployment.

  In the Service Deployment phase, WebSphere Service Registry and Repository is used for its role in storing, managing, and governing the operational service metadata that is required by connectivity and management scenarios.

  In the Service Management phase, WebSphere Service Registry and Repository is used to offer a more complete and declared definition of the service to compare against the observed definition of the service.

Repositories that have not been covered as part of this product identification in Table 3-1 on page 82 are considered out of scope of this publication. The repositories not covered include:

- Service Deployment - Other service endpoint registries/repositories

  Includes UDDI directories, external repositories and registries, and information-based services.

Part 2

Solution implementation

4

Our business scenario

This chapter introduces the case study used within this book. This chapter contains the following topics:

- Business scenario overview
- Business objectives
- Business model
- Business scenario requirements

4.1 Business scenario overview

As shown in Figure 4-1, the business scenario used in this IBM Redbooks publication is a simplified field work management scenario for a fictitious full service telecommunications provider, *ITSO-Tel*. In a typical business-to-consumer model, customers go to the retail unit to place orders for fixed line telephone, cellular telephone, broadband Internet, or cable TV services. Partner service providers go to the wholesale unit to place service orders for their customers.

The retail and wholesale systems request fulfilment of a consumer's order from the internal operations support systems, which respond as to whether field work is required. If field work is required, a field work appointment must be scheduled that is convenient for the customer.

The appointment and required tasks normally must be scheduled with the internal workforce. However, during peak sales times (such as cable TV sales during the buildup to the Australian Football League Grand Final[1] in Melbourne in September each year), the required additional demand needs to be met by external workforce providers using the business-to-business model.

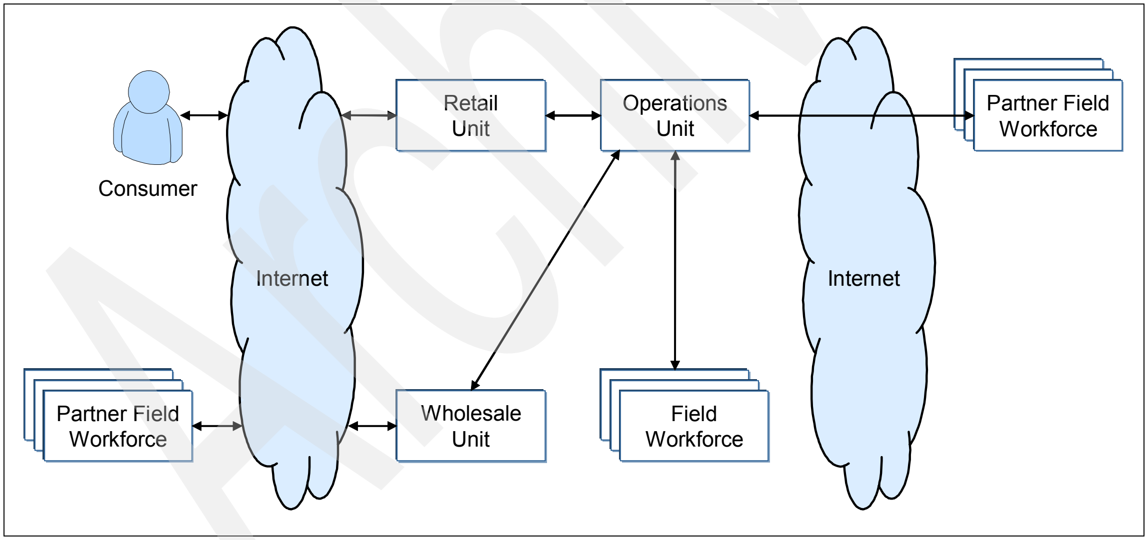


Figure 4-1 High-level business context

[1] A major Australian sporting event with a similar national impact to the NFL Super Bowl in USA or the Champions League Final in Europe.

4.2 Business objectives

This business scenario presents a number of challenges and opportunities for ITSO-Tel.

4.2.1 Business challenges

The business challenges presented by our scenario include:

- Allowing bundling discounts when new services are ordered is a critical capability for competing in a telecommunications market.
- Aligning customer appointments to minimize the number of visits required to provision multiple telecommunications services at a customer's premises.
- To remain competitive, the organization needs to move away from assisted channels, such as shop front or call center, toward self-service channels via the Internet or 3G mobile.
- Allowing customers to select a convenient appointment. Both retail and wholesale customers need to participate directly in the appointment booking process.
- The workforce-related service levels that need to be provided vary depending on the customer category. Customer categories include business customers and private customers. Special service levels with regulatory requirements also apply to some customers, such as rural customers or medical priority customers.

  Gold, Silver, and Bronze service levels should also be available to customers on a Fee for Service basis.
- Business policy requires that seasonal peaks in demand for field work activities are managed through the use of external partner workforces. These partner workforces potentially change with the contract expiration and renewal cycle.
- Multiple parties are involved as both providers and consumers of field workforce services. The retail and wholesale units act as consumers of the workforce services. There are multiple internal and external field workforce groups that each support a subset of the available telecommunications services. The capabilities of each of the workforce service providers are not standardized.

4.2.2 Business drivers and goals

Based on the challenges identified in 4.2.1, "Business challenges" on page 89, a number of business drivers and goals can be identified. Some examples are shown in Table 4-1.

Table 4-1 Business drivers and goals

Driver	Goal
Improve offerings' competitiveness	Support multi-play customer orders with bundling discounts applied.
Reduce costs	Avoid multiple service calls (visits): ► Consolidate the appointments for the individual service orders into a single appointment, where a field representative is involved.
Increase customer self-service	Offer valued capabilities: ► Offer capabilities that are important to the customer (such as self-selection of a convenient appointment time) through multiple self-service channels.
Improve business controls and regulatory compliance	Apply business policy to workforce unit selections: ► Enable key business policy decisions to be considered when selecting a workforce to complete field work. Allow use of external workforces when service level requirements cannot be met by the internal workforce due to seasonal peaks in demand.
Standardization of capabilities required by the organization	Standardize workforce capabilities across workforce units: ► Specify a standard set of capabilities to be delivered by all workforce providers.

4.3 Business model

The business model for ITSO-Tel is based on the TM Forum enhanced Telecom Operations Map (eTOM) process framework and Shared Information/Data (SID) model. These models define and structure the functional components and information needed by an organization to operate in the telecommunications industry.

4.3.1 Business component model

The TM Forum eTOM framework is a widely accepted standard for business processes in the telecommunications industry and has been adopted by ITSO-Tel. eTOM covers the full range of business processes that telecommunications service providers need to implement.

Business domains within the organization are based on the eTOM Level 1 horizontal and vertical process groups. As highlighted in Figure 4-2, our scenario lies in the Fulfillment and the Operations Support and Readiness domains, spanning the Customer Relationship Management (CRM), Service, Resource, and Supplier/Partner Management horizontal process groups.

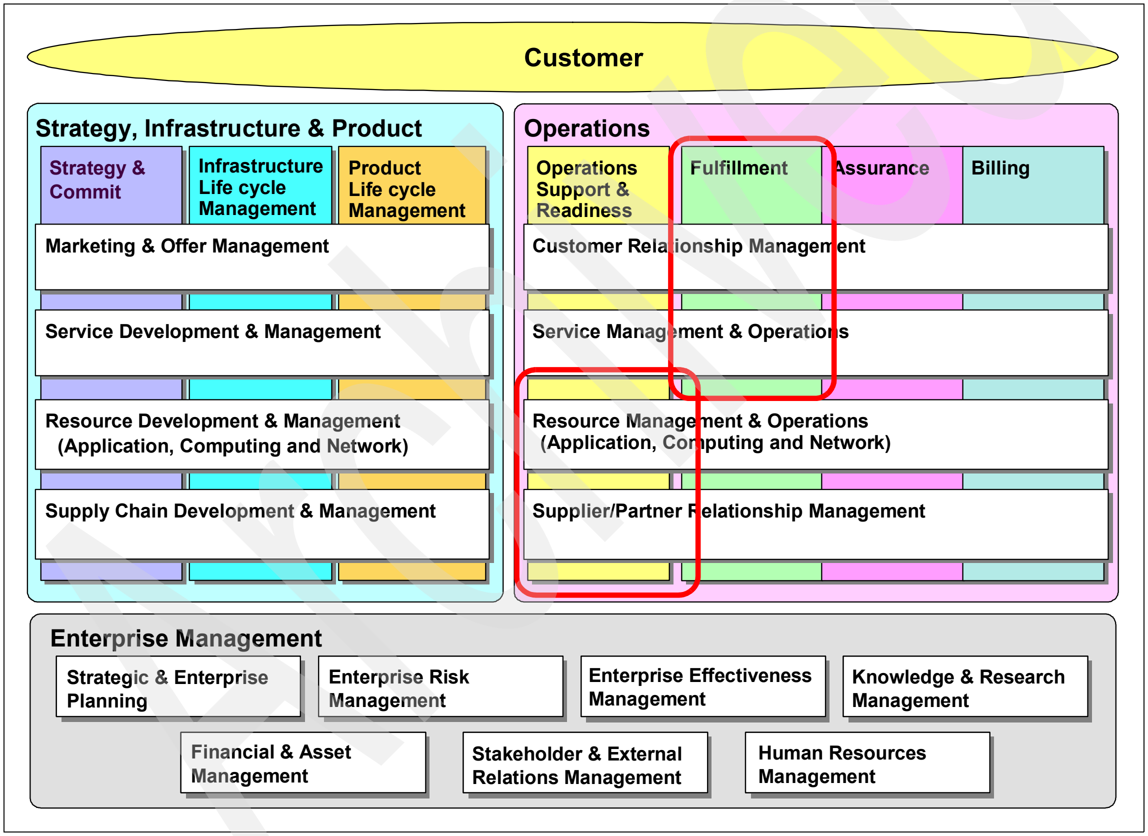


Figure 4-2 TM Forum enhanced Telecom Operations Map (eTOM)

Using the business process framework provided by eTOM benefits a telecommunications organization in a number of ways:

- It provides an industry standard framework for defining high level project scope and functional requirements across the organization and with suppliers and partners.
- It can be used to assign governance and delivery responsibilities to teams and individuals, from both the business and the IT perspective. As we saw in Figure 4-2 on page 91, eTOM places providers of workforce activities in the Operations Support and Readiness process group in the Resource and Supplier/Partner Management layers. For our scenario, consumers are in the Fulfillment process group in the CRM and Service Management layers.
- It is also used in the analysis of existing business processes and the design of new business processes. It provides a complete, enterprise-wide framework or catalog of processes that can be used to identify gaps in process coverage, or overlapping or duplicated coverage.

The process model presented in 4.3.3, "Process model" on page 94 was defined in this way.

For other industries or scenarios, there are a number of standard models available with similar goals. Examples include IT Service Management (ITIL) from the British Office of Government Commerce for Information Technology management and the Supply-Chain Operations Reference (SCOR) model from the Supply-Chain Council for supply-chain management.

4.3.2 Enterprise information model

ITSO-Tel has adopted the TM Forum Shared Information/Data (SID) model, an accepted standard information model for the telecommunications industry. For a retail industry example, you can use the Association for Retail Technology Standards (ARTS) Standard Relational Data Model in a similar way.

The SID model and eTOM in combination provide a standard way of describing how data and process need to fit together to meet a given telecommunications business need. eTOM defines the appropriate business processes, and SID defines the required information entities. The data and information artifacts required for a business process are sourced from a common data model, defined in the SID model.

Figure 4-3 on page 93 shows the SID domains and Level 1 Aggregate Business Entities (ABEs). ABEs are cohesive, multi-level groupings of business entities, similar in concept to the Level 1, 2, and 3 process groupings in eTOM.

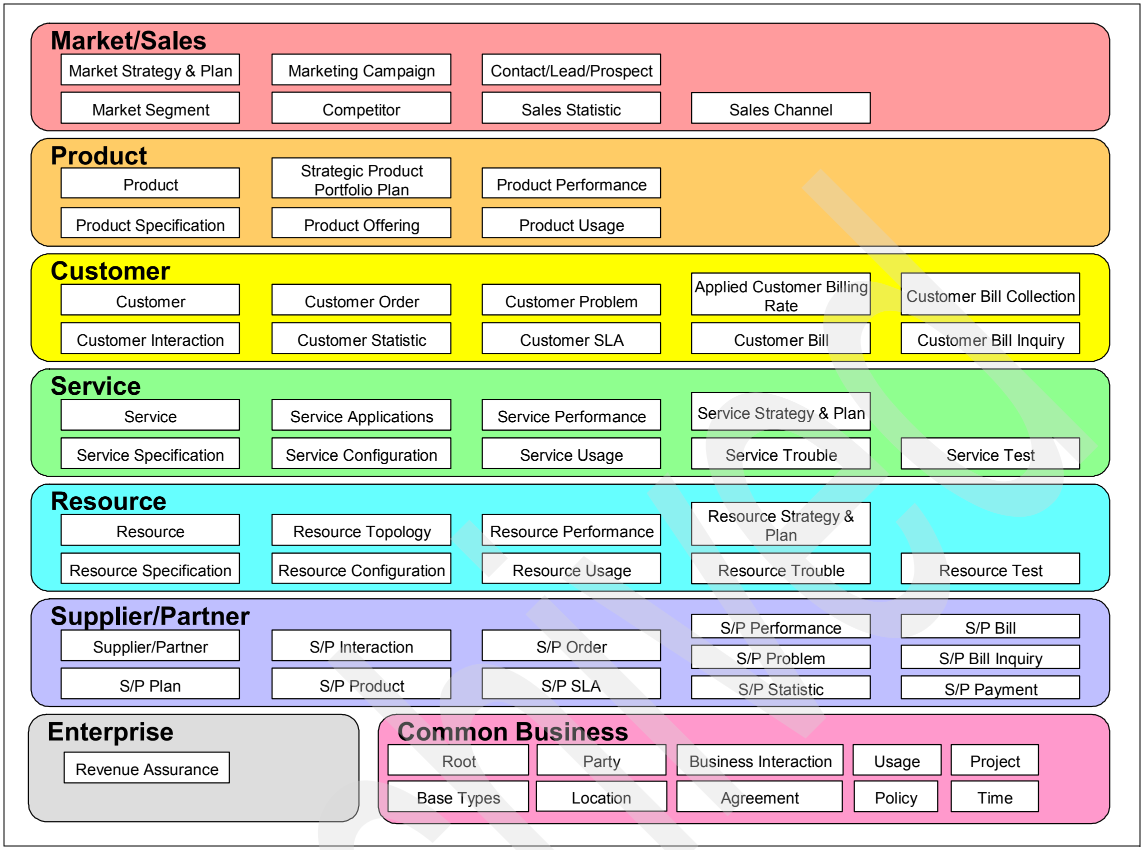


Figure 4-3 SID domains and Level 1 ABEs

The key ABEs of interest in our scenario include:

- From the Customer domain:
  - Customer Order detailing which products or services are required by the customer, for example, broadband cable service with modem
  - Customer service level agreement (SLA) for provisioning milestones
- From the Product domain:
  - Product Order detailing the required product and features, for example, cable modem model
- From the Service domain:
  - Service Order detailing the required service and features, for example, cable TV channels package, broadband cable speed, quotas, and so on

- From the Common Business domain:
  - Project for field work order or task details, for example, work estimate and task dependencies
  - Location for site details, for example, customer's service address
  - Base Types for start and end times, for example, appointment details
  - Party for organization details or skills, for example, company details where applicable or skills required for cable installation
- Entities, such as a Medical Priority indicator can require organization-specific extensions to the SID. This indicator is needed to highlight customers to whom ITSO-Tel has regulatory obligations to provide special assistance.

The entities identified should provide the common information catalog for the workforce-related business transactions in our scenario.

Refer to the TM Forum publication, *GB922 Business View Concepts, Principles and Domains* for an overview of the SID model. This publication is available at:

http://www.tmforum.org/library

4.3.3 Process model

The eTOM Operations horizontal process groupings of CRM, Service Management, Resource Management, and Supplier/Partner Management provide convenient paths for the major actors in our scenario. The CRM layer supports the front-office interactions with the customer. The Service and Resource Management layers model back-office capabilities in Operational Support Systems. The Supplier/Partner layer supports interactions with external business partners.

The TM Forum has also published a decomposition of the eTOM process framework down to level 3 activities in *GB921D Addendum D: Process Decomposition and Descriptions*. This publication is available at:

http://www.tmforum.org/library

You can use these activities to define a target business process across the horizontal eTOM process groupings.

Figure 4-4 on page 95 highlights key activities in the service provisioning process for our workforce scenario. Each activity shows the eTOM Level 3 Process Identifier and Name from the TM Forum GB921D addendum.

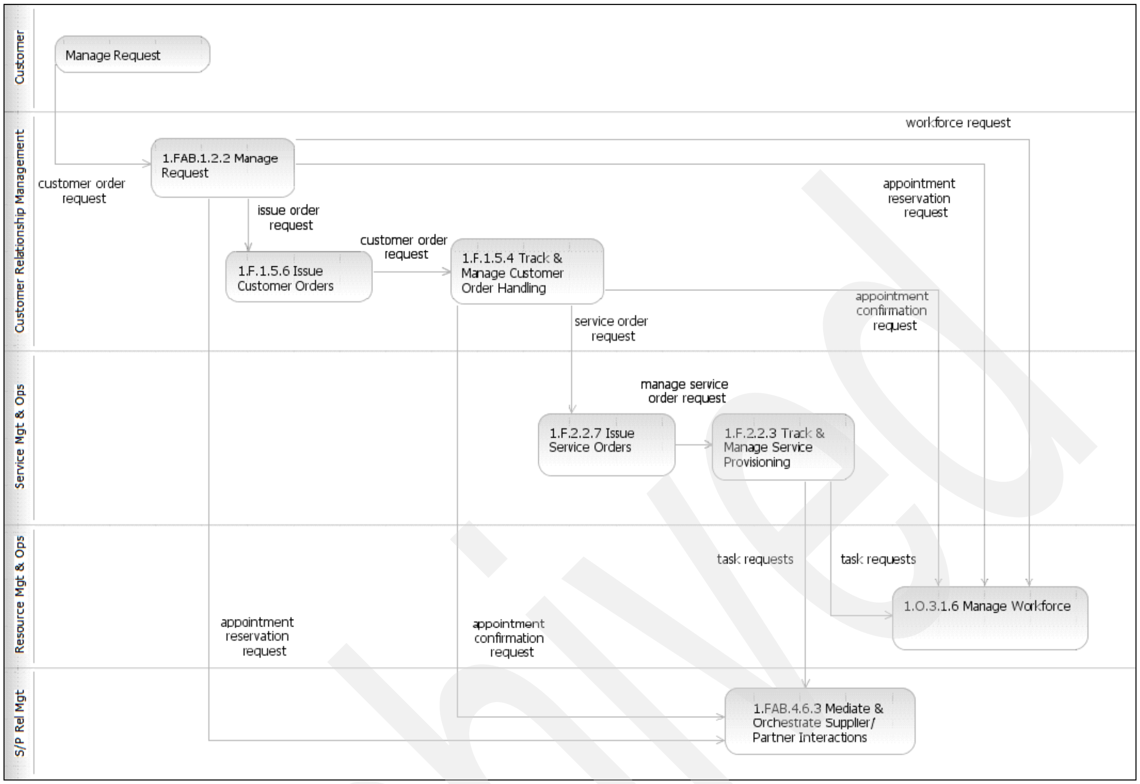


Figure 4-4 Scenario business process model

The sequence of eTOM level 3 activities in our business process requiring field work is:

1. A customer contacts the organization to request a new cable TV service and broadband cable service through one of the following channels:
   - CSR-assisted telephone call
   - Self-service Web or 3G mobile access
   - Wholesale agent

   The customer request is processed by the Manage Request activity.
2. The *Manage Request* activity determines that field work requiring a customer appointment is needed. To facilitate agreement on a convenient appointment time for the customer, Manage Request needs to provide a selection of available appointment slots from Manage Workforce (or from Mediate & Orchestrate Supplier/Partner Interactions for external workforces) then reserve the agreed upon slot.

For simple products and services, the Manage Request activity can complete a full workforce request, confirming the customer appointment and specifying the task details for the appointment in one step.

Where the customer request is for multiple services, the Manage Workforce activity needs to determine if the customer order can be completed in a single customer appointment. Every effort needs to be made to avoid multiple visits that are inconvenient for the customer and an inefficient use of field resources.

After determining the requirement for a customer appointment, the Manage Request activity determines that Customer Orders need to be issued and invokes the Issue Customer Orders activity providing the customer appointment details.

3. The *Issue Customer Orders* activity issues the customer order and invokes the Track and Manage Customer Order Handling activity to fulfill the order.
4. The *Track and Manage Customer Order Handling* activity orchestrates the fulfillment of the customer order. It will invoke the Issue Service Orders activity to issue individual service orders for the cable TV service and the broadband cable service. It will also confirm the customer appointment with Manage Workforce (or from Mediate & Orchestrate Supplier/Partner Interactions for external workforces) when the service orders have been issued.
5. The *Issue Service Orders* activity issues the service order and invokes the Track and Manage Service Provisioning activity to fulfill the service order.
6. The *Track and Manage Service Provisioning* activity orchestrates the fulfillment of each service order.

   Track and Manage Service Provisioning will invoke the required service design activities (not shown) to determine which tasks are required to provision the service. After this is complete, it will update the Manage Workforce activity with the service order and task details for the customer's appointment.
7. The *Manage Workforce* activity (or Mediate & Orchestrate Supplier/Partner Interactions for external workforces) is responsible for orchestrating the required field workforce service provisioning activities. These activities need to be completed in the negotiated customer appointment slot and in accordance with the appropriate Customer Service Level Agreement.

Note: This process model is intentionally incomplete. It is only meant to highlight points of interest for further exploration in our case study. For example, billing and status updates are not shown in our process.

4.4 Business scenario requirements

This section introduces some of the field work related functional and nonfunctional requirements for the ITSO-Tel scenario. The following actors are involved in field work appointments:

- Retail Unit for field appointments for ITSO-Tel customers
- Wholesale Unit for field appointments for ITSO-Tel wholesaler customers
- External Workforces for field appointments with business partner workforces

> **Note:** The requirements discussed here are intentionally incomplete. They are only meant to highlight points of interest for further exploration in our case study.

4.4.1 Functional requirements

Functional requirements describe the capabilities or behavior required by business stakeholders. They allow all stakeholders to understand every aspect of how a business capability works.

Table 4-2 provides examples of functional requirements for our field work management scenario.

Table 4-2 Business functional requirements

ID	Requirement
Driver goal	**Reduce costs.** **Avoid multiple visits to a customer.**
FR-01	Minimize the number of workforce visits to the customer's premises for multi-play customer orders: ► Ensure that a workforce only provides one appointment slot for all the service orders that they support ► Use a workforce that has the skills and capabilities to complete all required service orders in one appointment slot
Driver goal	**Increase customer self-service.** **Offer valued capabilities.**
FR-02	Schedule and cancel appointments for field work at the customer's premises.

ID	Requirement
FR-03	Support appointments for the following products and services: ► Fixed line telephone ► ADSL broadband ► Cable broadband ► Cable TV
FR-04	Select an appointment slot from a choice of 10 available appointment slots.
Driver goal	**Improve business controls and regulatory compliance. Apply business policy to workforce unit selections.**
FR-05	Schedule field work with both internal and external field workforces.
FR-06	Enable business policy changes to drive the selection of the internal or external workforce unit to use.
FR-07	Select the appropriate workforce unit to be based on current business policy: ► Internal workforce units will be used in preference to external workforce units. ► Workforce unit's current product and service skills or capabilities ► Workforce unit's current ability to meet the appointment SLA
Driver goal	**Standardization of capabilities required by the organization. Standardize workforce capabilities across workforce units.**
FR-08	Manage all field workforce appointment activities through a single business function.
FR-09	Link all provided business functions to eTOM level 3 process descriptions.
FR-10	Link all provided business information to SID entities at the leaf level or leaf extension level.

4.4.2 Nonfunctional requirements and SLAs

Nonfunctional requirements define the quality of service that is required from the business system. They include properties or characteristics of the business system about which its stakeholders care and hence affects their degree of satisfaction with the system.

Common categories of quality of service are:

- Availability
- Performance

- Security
- Standards
- Usage

Table 4-3 provides examples of nonfunctional requirements for our field work management scenario.

Table 4-3 Business nonfunctional requirements

ID	Requirement
Category	**Performance**
NFR-01	Transaction response times: ► Appointment reservation 6 seconds average 95% less than 10 seconds ► Appointment confirmation 3 seconds average 95% less than 5 seconds
NFR-02	Transaction throughput: ► Appointment reservation 300,000 transactions per day 150,000 transactions per hour at peak hour ► Appointment confirmation 200,000 transactions per day 100,000 transactions per hour at peak hour
Category	**Availability**
NFR-03	Available hours: ► 24x7
NFR-04	Percent uptime: ► Percent uptime 99.99%
Category	**Security**
NFR-05	Conform to ITSO-Tel security policy
Category	**Standards**
NFR-06	Conform to ITSO-Tel Business, IT, and SOA governance policy

ID	Requirement
Category	**Usage**
NFR-07	Support required Customer Service Level Agreements: ► Medical Priority ► Corporate ► Metropolitan ► Regional ► Fee for Service (Gold, Silver, and Bronze)

A *Service Level Agreement* (SLA) is an agreement or contract between the provider of a business service and the consumer. It sets expectations for the level of service with respect to availability, performance, and other measurable objectives.

Define SLAs with specific metrics and methods for monitoring and analyzing the adherence to the service level objective. The definition of service level agreements and verification of compliance is a key activity of the Service Level Management process.

Table 4-4 provides Customer SLA examples for our field work management scenario.

Table 4-4 Service level agreements

ID	Service level agreement
Category	**Medical Priority**
SLA-01	On-site within 4 hours, time 24x7
Category	**Corporate, Gold**
SLA-02	On-site within 5 days, time 9 am to 5 pm Monday to Friday
Category	**Metropolitan, Silver**
SLA-03	On-site within 10 days, time 9 am to 5 pm Monday to Friday
Category	**Regional, Bronze**
SLA-04	On-site within 15 days, time 9 am to 5 pm Monday to Friday

5

Service identification, specification, and realization

Effective service-oriented architecture (SOA) governance is key to achieving the true value from SOA. This chapter discusses the application of governance within the Model life cycle phase of an SOA project.

This chapter applies the discussion of governance to the fictional ITSO-Tel organization, defined in Chapter 4, “Our business scenario” on page 87.

This chapter contains the following sections:

- Introduction
- Service identification
- Service specification
- Service realization decisions

5.1 Introduction

The objectives of this chapter are to:

- Highlight the management and governance considerations during the service life cycle progression between the business and IT domains.
- Demonstrate the progression of a service from the requirements phase through the specification phase using the scenario.

5.1.1 Approach

The approach taken in this chapter is to:

- Utilize three phases of service life cycle progression for the Model phase:
  - Service identification
  - Service specification
  - Service realization
- For each service life cycle phase, identify:
  - Activities that facilitate the completion of those phases
  - Governance decisions applied to the activities
- Apply the service life cycle phases, activities, and governance decisions to the ITSO-Tel business scenario.

5.1.2 Assumptions

Throughout this book, we use a scenario based on the hypothetical telecommunications provider ITSO-Tel to illustrate key concepts and activities.

In this scenario, we assume the following preconditions, which are based on earlier chapters, have been met:

- An SOA Governance and Management capability exists as outlined in Chapter 2, "Establishing an SOA governance and management capability" on page 21.
- The business scenario to be addressed as outlined in Chapter 4, "Our business scenario" on page 87

Additionally, we make the following assumptions:

- The corporation has an existing SOA runtime capability.
- The corporation has an existing Manage Internal Workforce business service that manages its internal workforce scheduling for telecommunication products.
- The corporation is concerned with governance of its assets. It has chosen to utilize Rational Asset Manager as a repository and life cycle management system.
- The corporation has an IT inventory of existing operational systems and services within its asset repository.

5.1.3 SOA governance goals

The goal of SOA governance is twofold:

- Determining and defining roles and responsibilities of those involved in the identification and delivery of services
- Establishing policies, controls, and measurements that enable people to fulfill their roles and responsibilities

We address the best way to establish roles and responsibilities for SOA delivery in Chapter 2, “Establishing an SOA governance and management capability” on page 21. In this chapter, we address the establishment and implementation of policies and controls with regard to the analysis and design of SOA services. With the establishment of sufficient governance, you can ensure that needed assets are available for services to progress through the next phase of their life cycle.

Asset governance

The primary means of providing governance is through the management of various work products and assets relating to the services.

The governance policy for these assets must address:

- Ownership

  Each asset must have a specified owner, reviewers, and stakeholders.

- Availability

  Assets must be stored in a repository that makes current and prior versions of the documents available to appropriate parties.

- Association

  The dependencies between services and assets need to be tracked.

- Life cycle

  Each asset type must have a defined life cycle describing its creation, approval, and change management.

- Measurement

  You must have the ability to determine which assets and services are dependent on missing assets or approvals to proceed on their life cycle. Additionally, you must identify the processes and life cycles that have proceeded despite policies or requirements not having been met.

Beyond control and management of assets relating to services, governance provides value in progressing services through the three phases of SOA modeling: Service Identification, Service Specification, and Service Realization decisions (Figure 5-1).

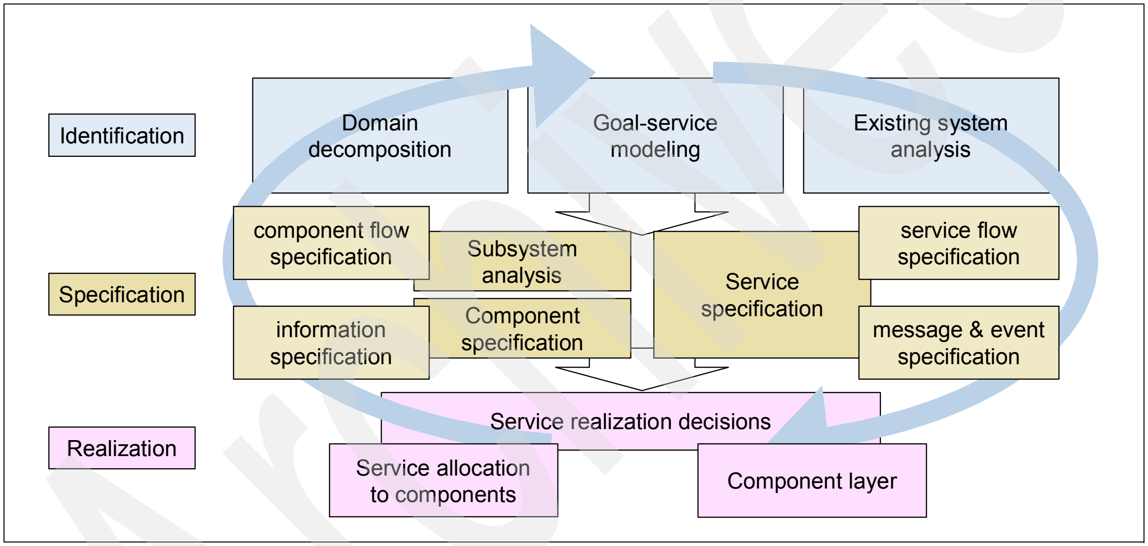


Figure 5-1 Service-oriented Modeling and Analysis method

You can read more information about Service-oriented Modeling and Architecture on developerWorks® at:

`http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/`

You can read information about IBM Rational Unified Process (RUP) for Service-oriented Modeling and Architecture V2.4 at:

`http://www.ibm.com/developerworks/rational/downloads/06/rmc_soma/`

5.2 Service identification

Service identification involves the following techniques:

- Domain decomposition
- Goal service modeling
- Existing asset analysis

Each of these techniques creates and relies upon a variety of assets. You can enact governance by establishing mechanisms to control and monitor these assets. These assets will also provide input and guidance for governance further along in service specification and realization (Figure 5-2).

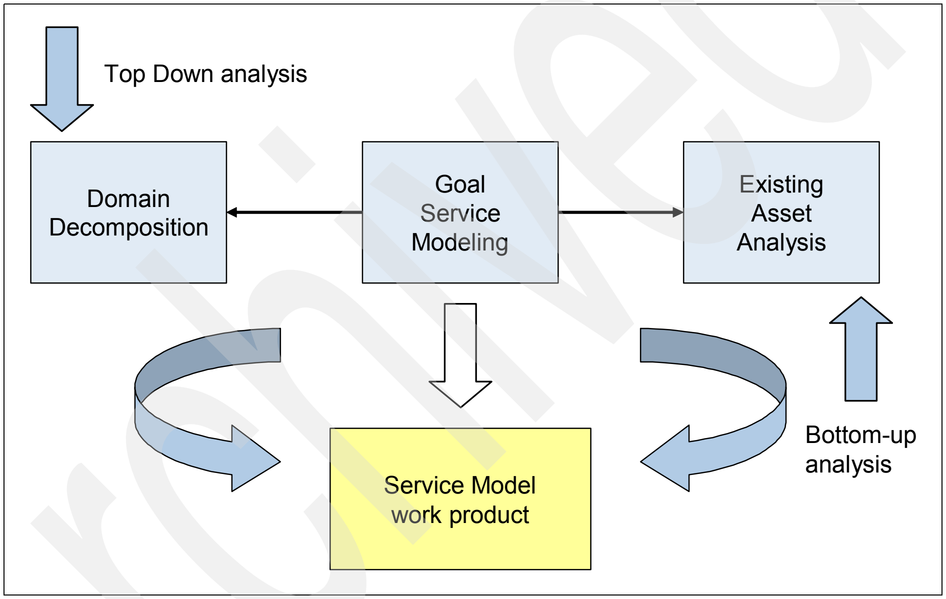


Figure 5-2 Service identification

In Table 5-1 on page 106, we introduce the major techniques, activities, and key governance decisions required for each technique.

Table 5-1 Service identification governance

Techniques	Activities	Governance decisions/checklist
Domain decomposition	► Identify candidate services using a top-down business-driven decomposition. ► Identify boundaries for subsystem business processes flows, commonality, and variations of functionality. ► Derive from Functional Area Analysis, Business Process Model, Use Case Model, and Variations Model.	► Are required documents available, complete, and current? ► Do required documents leverage the industry standard for Business process models? ► Do required documents leverage industry and corporate standard modeling notation and practice? ► Is the candidate business process flow defined at a high level?
Existing asset analysis	► Examine assets, such as existing custom applications, packaged applications, and industry models to determine what can be leveraged to realize service functionality. ► Uncover any services that might have been missed through process decomposition and goal service modeling.	► Are existing assets identified with a functional granularity similar to others business services identified in the Service Model? ► Have candidate services, flows, and packaged APIs been validated to determine what can be leveraged?
Goal service modeling	► Goal service modeling ensures that key services have not been missed in the top-down or bottom-up analysis. Decompose business goals into sub-goals. ► Identify candidate services to meet sub-goals. ► Identify key performance indicators and associated metrics that allow for the attainment of the sub-goals through the services identified.	► Are abstract business goals broken down into concrete sub-goals that allow them to be fulfilled? ► Are all goals and sub-goals addressed by candidate services? ► Have Domain experts signed off on the Service Model that the Repository broadcast to them as “awaiting sign off”?
General SOA governance	N/A	► Is the Service Model committed to the repository? ► Are the candidate Services traceable to business goals?

5.2.1 Service identification scenario

Throughout this book, we use a scenario based on the hypothetical telecommunication provider ITSO-Tel to illustrate key concepts and activities.

5.2.2 Domain decomposition

During scenario domain decomposition for ITSO-Tel, the following steps took place:

1. The following business assets were retrieved from the repository:
   - Business Component Model

     A breakdown of the business based on the eTOM framework
   - Process Model

     A high-level description of business processes derived from eTOM level 3 processes. Assessing the eTOM Process model, identified relationships between actors in the Fulfillment domain and providers in the Resource Management Operations domain with regard to workforce management
   - Data Model

     A data model describing the structure of and relationships between pieces of information in the enterprise. This data model is based on the TM Forum NGOSS Shared Information Data model (SID).
   - Service Model

     A collection of services and service components in various stages of development from candidate through deployment and retirement. Included in the Service Model are the specification and realization decisions for exposed services.
2. The Resource Management Operations domain was found to contain a Manage Request business process. This business process was found to contain sub-processes Manage Workforce and Mediate and Orchestrate Supplier/Partner Interaction.
3. The existing Service Model was found to relate the Manage Workforce business process to an existing Manage Internal Workforce service.

5.2.3 Existing asset analysis

Subsequently, we identified two relevant assets by consulting the IT inventory within the repository:

- The Manage Internal Workforce service (also found in the domain decomposition)
- An existing operational system interface providing for management of the appropriate external workforce

5.2.4 Goal service modeling

Undertaking goal service modeling in the Manage Workforce business process revealed:

- The business goals stated the need to:
  - Improve business controls and regulatory compliance by applying business policy to dynamically determine workforce type (internal as opposed to external) selection.
  - Standardize the capabilities of the organization across the workforce units.
  - Increase customer self-service by allowing self-selection of workforce appointment time through multiple channels.
- It was determined that a single business service that managed both internal and external workforces was the optimum future implementation candidate service to meet these goals.
- The Service Model was expanded to add the business description of the new candidate service Manage Workforce to the Service Portfolio and Service Hierarchy.
- The Service Model was submitted into the repository for review.

5.2.5 Results

The results of this analysis were captured within the Service Model work product, which contained updated Service Portfolio and Service Hierarchy sections.

5.3 Service specification

Service specification involves the following techniques:

- Service specification
- Subsystem analysis
- Component specification

These techniques are used to determine which services to expose to business process flows and consumers. Furthermore, it elaborates on any internal flows and service components that are necessary to fulfill the service.

In Table 5-2 on page 109, we introduce the major techniques, the activities, and key governance decisions that are required for each technique.

Table 5-2 Service specification governance

Techniques	Activities	Governance decisions/checklist
Service specification	► Decide which possible candidate services need to be developed into exposed services. ► Determine service dependencies, composition, and service flows. ► Determine nonfunctional requirements. ► Specify service message models and content. ► Specify state management.	► Do all services for assembly pass the service litmus test? ► Were the service litmus tests approved by the SOA Governance council? ► Was the Service Model updated with analysis and findings? ► Are components identified that are appropriate to be further developed? ► Were all dependent services and Operational Systems identified and in the IT Inventory of the repository? ► Is there traceability to the requirements? ► Are all message elements defined in terms of enterprise data models? ► Are all message elements validated as interoperable? ► Have state management considerations been documented?
Subsystem analysis	► Identify service component containers that will realize the services. ► Service technical specifications.	► Have identified subsystem owners confirmed system availability and capacity? ► Have technical specifications been submitted to the repository? ► Have technical specifications that the repository broadcast as “awaiting sign off” been signed off on by domain experts?
Component specification	► Define the details of service components.	► Is formalized notation used to reflect components?
General SOA governance	N/A	► Is the Service Model committed to the repository? ► Are documentation and interface artifacts related to the Service Model in the repository?

5.3.1 Service specification scenario

In this scenario, we presume that we have completed the Service Identification process for the Manage Workforce scenario presented. Our governance checklist must ensure that we have appropriate assets to proceed with service specification.

5.3.2 Service specification

Beginning the service specification process, we further investigated the Manage Workforce candidate service. It was found to have the following characteristics indicating that it is business-aligned:

- The service directly supports the Manage Workforce business process.
- There is business willingness to fund the service throughout its life cycle.
- The business owner intends to share the service across the enterprise.

It was found to have the following characteristics, which indicated that it is suitable for composition:

- The Service meets identifiable business quality of service attributes.
- The service is stateless in nature.
- The service is self-contained and can be deployed independently to meet its business goal.
- The service is technology-neutral and can be deployed via a standardized interface technology.

It was found to have the following characteristics, which indicated it is suitable for externalization:

- The service has an externalized service description.
- The service is self-sufficient and does not rely on any further information in order to describe its message exchange pattern or data types.

It was found to have the following characteristics, which indicated that any redundancy had been prevented:

- The service can be leveraged by additional business process consumers other than the Manage Workforce process.

These characteristics suggested that the candidate service is appropriate for externalization.

The Manage Workforce service was further elaborated on during this phase as follows:

- The service was determined to be dependent on and composed of:
  - The existing Manage Internal Workforce service
  - The existing operational interface for Manage External Workforce
- The service is constrained by the following business nonfunctional requirements:
  - NFR-01 Response time
  - NFR-02 Volumetric
  - NFR-03 Availability
  - NFR-05 Security
  - NFR-06 Governance
  - NFR-07 Channel of usage
- The Service was described with the message specification shown in Table 5-3.

Table 5-3 Message specification for ManageWorkforce

Topic	Input	Output
GetAppointment	AppointmentRequest	AppointmentConfirmation
CancelAppointment	AppointmentConfirmation	BooleanConfirmation

- A Manage Workforce WSDL file was created detailing these operations and message structures in accordance with Enterprise Data Model constraints.
- The service was validated as not containing a functional state.

5.3.3 Subsystem analysis

The Manage Workforce service was found to require the IT subsystem execution dependencies shown in Table 5-4 on page 112.

Table 5-4 Subsystem analysis for Manage Workforce

Layer	Scenario service name	Subsystem analysis
Service	Manage Workforce	New
Service components	Manage Workforce	New
	Manage Internal Workforce	Existing service component
Operational systems	Manage Internal Workforce	Existing custom application
	Manage External Workforce	Existing packaged application API

Figure 5-3 is presented for the context of the intended abstract services layered architecture.

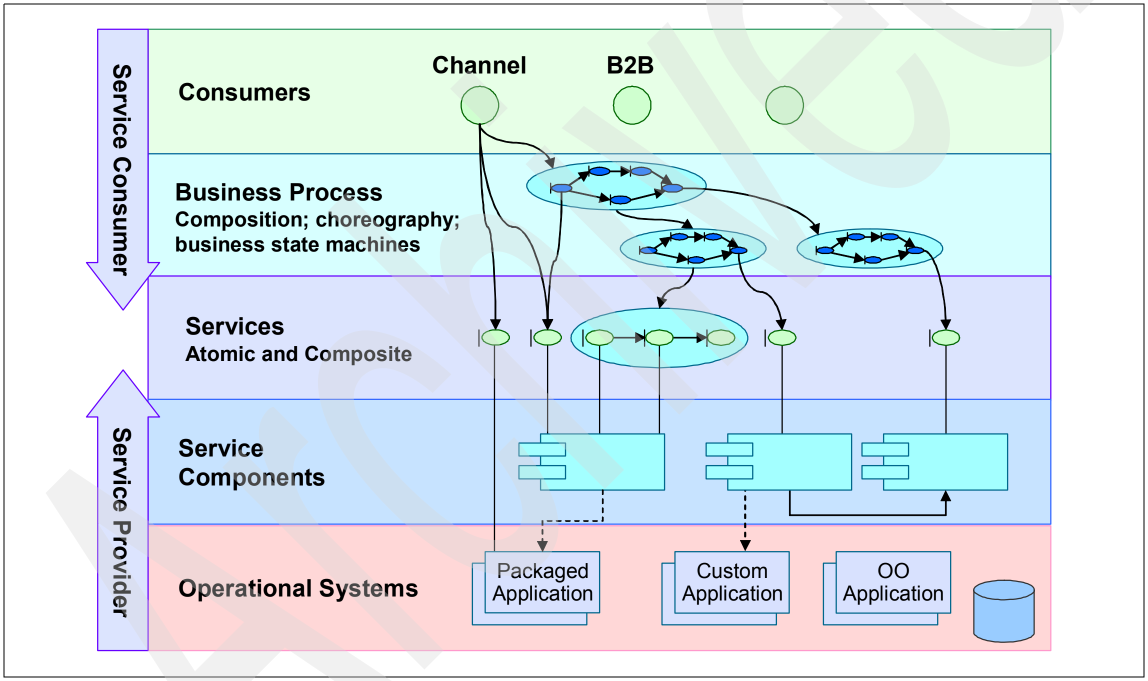


Figure 5-3 SOA Foundation architecture layers

5.3.4 Component specification

The Manage Workforce service was expanded via the following techniques in this phase:

- Component Sequence diagram
- Component Class diagram
- Variation Orientation Design

During the Variation Orientation Design, we identified the following points of variability in the Manage Workforce service:

- Appointment Customer
- Appointment Products
- Appointment Date
- Appointment Address
- Date
- Channel
- User

It was determined that for the desired functionality of dynamically determining workforce type that the Appointment Date and Appointment Address provided the key business policy context.

5.3.5 Results

The results of this analysis and specification were captured within the Service Model work product, which contained updated Service Exposure, Service Dependency, Service Composition, Service NFR, Service Message, and State Management sections.

5.4 Service realization decisions

Service realization involves the following techniques:

- Service allocation
- Component allocation to layers
- Technical feasibility exploration

After services and service components have been identified and specified, architectural and design decisions are made about exactly how each component will realize their functionality. Realization-related decisions are made and iterated throughout the life cycle and can occur in parallel with other activities.

In Table 5-5, we introduce the major techniques, the activities, and key governance decisions required for each technique.

Table 5-5 Service realization decisions governance

Techniques	Activities	Governance decisions/checklist
Service allocation	► Map services to components. (Note: Service allocation is an ongoing process during the design phase. Iteration occurs as the understanding of services and components increases).	► Have new component containers been identified and versions ratified? ► Have Service functional and nonfunctional requirements been reviewed by component owners? ► Has infrastructure change impact been analyzed?
Component allocation to layers	► Component allocation to layers ► Component decoupling	► Is a Layered Mediation Architecture in place? ► Are its layer responsibilities known and governed? ► Are components aligned to layer responsibilities?
Technical feasibility exploration	► Technical constraint determination ► Technical prototype	► Have risks and outcomes been assessed by stakeholders? ► Has governance been updated with any changes brought about by new components?
General SOA governance	N/A	► Is the Service Model committed to the repository? ► Are all architecture decisions documented and signed off on?

5.4.1 Service realization decisions scenario

In this scenario, we presume that we have completed the Service Specification process for the workforce management scenario presented. Our governance checklist ensures that we have appropriate assets to proceed with realization decisions.

5.4.2 Service allocation

During service allocation, we determined that the Manage Workforce business service will be associated with the Manage Workforce dynamic execution policy component.

5.4.3 Component allocation to layers

This is the allocation of components to layers:

- The Manage Workforce service will be a new exposed business service via a messaging adapter component.
- The Manage Workforce service will utilize a policy layer adapter to determine business execution policy (such as whether an internal or external workforce will be utilized). Where policy determines an internal workforce is required, the existing Manage Internal Workforce will realize the service interface. Where an external workforce is required, the new Manage External Workforce API will be integrated to realize the service interface.
- The Manage Workforce service will utilize a semantic mediation layer to map from Enterprise Data Model message structure of the exposed business service to the Manage External Workforce existing operational system message structure.
- The External Workforce will be a new operational system outside the corporate firewall.
- The Internal Workforce Operational system will continue without functional change but will be deprecated.

5.4.4 Technical feasibility exploration

The technical feasibility exploration of the Manage Workforce service draws upon the results of existing asset analysis and confirms that the functional and nonfunctional requirements can be met. In our case, existing containers were validated as having the required functionality and capacity to achieve the intended business service goal. Special consideration was applied to the new external Manage External Workforce system. It was validated for its integration technology, message model, exception handling, data validation, security, and service level agreements (SLAs).

5.4.5 Results

The completed Service Model assets were stored and signed off on within the Rational Asset Manager repository. The Manage Workforce service specification was then eligible for entry into the assembly phase.

6

Service life cycle management using Rational Asset Manager

This chapter describes how Rational Asset Manager V7.0 can help establish service-oriented architecture (SOA) governance and an SOA asset repository. It also shows the integration between Rational Asset Manager and a runtime repository running in WebSphere Service Registry and Repository.

This chapter contains the following sections:

- Introduction
- Establishing governance in an asset repository
- Service discovery
- Service creation
- Service publication
- Evaluating asset repository usage

This chapter uses the fictional organization ITSO-Tel (described in Chapter 4, "Our business scenario" on page 87) to illustrate the use of Rational Asset Manager.

6.1 Introduction

Before the discussion can proceed about Rational Asset Manager, we need to provide a number of key concepts for background information and your understanding of the product. These include looking at Rational Asset Manager as far as:

- Defining the terms used by the product
- Contributing to the SOA life cycle
- Relating assets with a service

6.1.1 Glossary of terms

Rational Asset Manager provides a set of terms that require explanation to ensure that you understand to what each term refers. These are identified in Table 6-1.

Table 6-1 Rational Asset Manager glossary of terms

Rational Asset Manager term	Description
Asset	Assets are a collection of artifacts, or files, that provide a reusable solution to a specific business problem. To facilitate reuse, assets contain descriptive information that explains their purpose, use, and relationship to other assets.
Asset attribute	Rational Asset Manager has required, optional, and customizable information, called *attributes*, that users can include when submitting an asset to the repository. This information facilitates searching and helps other users make choices for how and when to use an asset.
Asset category	Defines the category for the asset based on the category schema and is associated with the asset. It is used for organizing assets so that users can find and reuse them.
Asset consumer	The asset consumer searches and browses reusable assets in the asset repository and ultimately uses the assets. The asset consumer provides feedback on the assets.
Asset producer	The asset producer updates an asset version by developing artifacts and harvesting artifacts as well as making changes to the metadata.

Rational Asset Manager term	Description
Asset relationships	Relationships enable users to specify which assets are related to each other within the repository and to describe the nature of that relationship. Relationships facilitate search, governance, and asset reuse.
Asset repository	The asset repository focuses on the development community. This is the storage point for reusable assets for the enterprise. It is used to manage assets, such as documents, code, processes, and binaries with the ability to search and retrieve assets for use.
Asset type	An asset type is the primary level of organization within the Rational Asset Manager repository. Asset types facilitate governance and searching. When users submit assets, the asset type controls the information and artifacts that users must include with the asset. Repository administrators create asset types, which include the asset type name and description.
Category schema	Category schemas organize assets within the repository so that users can find and reuse them. A category schema consists of a group of categories. Repository administrators create category schemas based on the needs of their company, its structure, and the types of assets that their company creates.
Community	A *community* is a collection of assets grouped by a common use and purpose. The assets within a community can have different types, categories, relationships, and review processes. Community administrators can assign roles and permissions to users and user groups in order to set different levels of access to the community.
Community administrator	Community administrators manage the access, review processes, and change management integration for groups of assets called communities.
Repository administrator	The repository administrator sets up the structure for how the repository is organized and how users interact with the repository. The repository administrator also manages communities and asset types, relationships, categories, and attributes.
Review process	Review processes exist within communities to ensure that an asset is complete, accurate, and optimized for reuse. Users or user groups can review assets according to their specific area of expertise.

6.1.2 Rational Asset Manager and SOA

Rational Asset Manager is the development asset management tool for use during the Model and Assemble phases of the SOA life cycle with a minor role in the Deploy phase.

The roles that Rational Asset Manager plays in each of the SOA life cycle phases include:

- Model

  Its role is the identification and discovery of business processes, requirements, and services that can be applied for the modeling of new or modified services.

- Assemble

  Discovery, retrieval, and reuse of existing services during the construction of business processes that have been reviewed and accepted in use in an organization. New services that have been developed will be stored in Rational Asset Manager to ensure that they are available for reuse within the organization, after passing through the SOA governance life cycle for the service.

- Deploy

  On review and acceptance of a service, initiate deployment to WebSphere Service Registry and Repository in a controlled manner for use in the runtime environment.

Rational Asset Manager is also the tool in ensuring the governance of SOA activities during the development part of the SOA life cycle and controlling of assets to service registries and repositories. This is achieved through:

- Providing a common repository for access to assets that will include services, documents for processes, and designs, as well as other artifacts
- Integration to Rational ClearQuest for the provision of defects and changes against assets allowing traceability and control of the assets
- Provision of a review process to ensure that assets are released for use only after key stakeholders have identified that they are of sufficient quality, satisfy requirements, and meet standards prior to public use in the organization
- Control of access to assets via access privileges and based on the role of the user accessing the resource
- Integration to WebSphere Services Registry and Repository to ensure the deployment of services is performed in a controlled manner based on the roles of the user

6.1.3 Services as assets in Rational Asset Manager

Rational Asset Manager is a storage repository for various types of assets and can be used for the support of the development process and the artifacts associated with it. It is supplied with a number of predefined “ready to use” *category schemas* that are commonly used in organizations. Rational Asset Manager provides the option for an organization to use these predefined category schemas, modify what is supplied, or create new category schemas based on the governance processes enforced. To assist readers in how services map to assets, we describe in detail:

- What is an asset in Rational Asset Manager
- Support for services in Rational Asset Manager

> **Note:** A *service* is one type of instantiation of an *asset*.

What is an asset in Rational Asset Manager

An *asset* is a collection of artifacts or files providing a reusable solution to specific business problems. Assets contain descriptive information that explains their purpose, use, and relationship to other assets. *Artifacts* are files that users can group together to form assets. Artifacts can be work products from software development processes, such as software requirements, designs, models, source code, data, tests, user interfaces, and documentation. The features of an asset include:

- Asset size

  Assets have no size limit. For example, a large asset can include all of the elements of a business requirement, such as its use cases, design models, components, component specifications, test cases, test drivers, and test data. An example of a small asset is an asset that contains information about a specific problem, such as test cases for an implementation.

- Metadata

  Assets contain descriptive information, or *metadata*, that enables you to find and reuse assets. Asset metadata includes information, such as name, owner, description, review state, and version. You can classify assets by type, category, relationships with other assets, tags, and attributes. You can also include artifacts that explain the goals, processes, and motivations for creating and using assets. Furthermore, administrators can configure the repository so that when users submit an asset, they are required to provide specific metadata.

- Asset reuse

  Assets can be reused in more than one context. One benefit of asset reuse is that it allows a company to avoid the costs associated with producing redundant assets in different branches. For example, a particular branch of a large company develops an implementation for a pop-up help window in a Web-based application. A different branch of the company also has a need for a pop-up help window. Instead of developing their own implementation, developers within the other branch use Rational Asset Manager to search for an asset that fits their needs. They find the preexisting asset implementation for a pop-up help window, download it, and modify it to suit their particular context.

- Review process

  Review processes help to ensure that an asset is complete, accurate, and optimized for reuse. Users or user groups can review assets according to their specific area of expertise.

- Governance

  Asset governance allows a company to control who can access, view, or modify an asset. Furthermore, a company can control how users modify assets and set requirements for users to include specific metadata or artifacts with an asset. For example, a tester needs to download an asset for testing purposes. A developer, however, needs to not only download the asset but also make changes to the asset. Governance allows users to interact with the repository in different ways according to task requirements.

- Asset states

  Assets can exist in several states:

  – Draft

    Asset is viewable only to the asset owner and administrators. The asset owner can make changes to the asset.

  – Review

    Asset is under review. Reviewers can participate in discussion forums about the asset.

  – Approved

    Asset has passed the review. The review board has the highest level of authority to approve an asset, followed by reviewers.

  – As is

    Asset bypasses review process. Community administrators can designate “as is” assets to require administrative approval.

  – Retired

    Asset remains available in the repository to asset owners or administrators, but cannot be downloaded.

– Deleted

  Asset is permanently removed from the repository.

Support for services in Rational Asset Manager

Rational Asset Manager provides a predefined, ready to use *category schema* for SOA for defining categories of the stored and managed assets. You use category schemas for categorization in Rational Asset Manager to ensure that you can search for and retrieve assets from the repository appropriately in order to successfully reuse the assets. The predefined, ready to use category schema for SOA is identified as *SOA* in Rational Asset Manager and is outlined by the following major categories and first level of sub-categories:

- Strategy

  Used for categorizing assets that encompass the strategy related to services including sub-categories of:

  – Reuse Strategy
  – Governance Strategy
  – Business Vision
  – Business Goal

- Reference Architecture

  Categorizes the types of reference architectures for the services, with each sub-category having more specific examples of each type. The main sub-categories are:

  – Operational Architecture
  – Deployment Architecture
  – Application Architecture

- Service Specification

  Identifies the categories used for specification of services with each sub-category having a hierarchy identifying more specific aspects. The sub-categories provided include:

  – Identification
  – Specification
  – Realization

- Reference Implementation

  The reference implementation is used for categorizing assets identifying the source of the reference implementation. The sub-categories provided include:

  – Production Quality
  – Non-Production Quality
  – Non-Commercial Use
  – Commercial Use

– Standard Implementation
– Extension Implementation

This category schema can be used as is by the organization in identification of service assets or modified, or a new category schema can be created specific to the organization needs and governance regime.

In addition, when adding assets to Rational Asset Manager, a number of asset types are available which associate the artifacts with an identified type for classification. Table 6-2 shows a list of the predefined asset types that are provided.

Table 6-2 Asset types used for service assets

Asset type	Description
Architecture	Architectural description to which service interfaces, implementation, db designs, and other assets must comply.
Business Process	Describes business functions and tasks. A business process can contain other business processes.
DB Design	Design of a database.
Service Design	Design of service interface or service implementation.
Service Impl	Implementation of a service.
Service Interface	Interface to one or more services.
Service Registry	Service Registry category.
Service Test	A test for a service interface or service implementation.
WSDL	WSDL that represents a link to WebSphere Registry and Repository.
XSD	XSD that represents a link to WebSphere Registry and Repository.

Excluding the last two asset types (WSDL and XSD, which relate to a link to WebSphere Registry and Repository), the attributes and associated relationships between the other asset types are shown as a Unified Model Language (UML) diagram in Figure 6-1 on page 125.

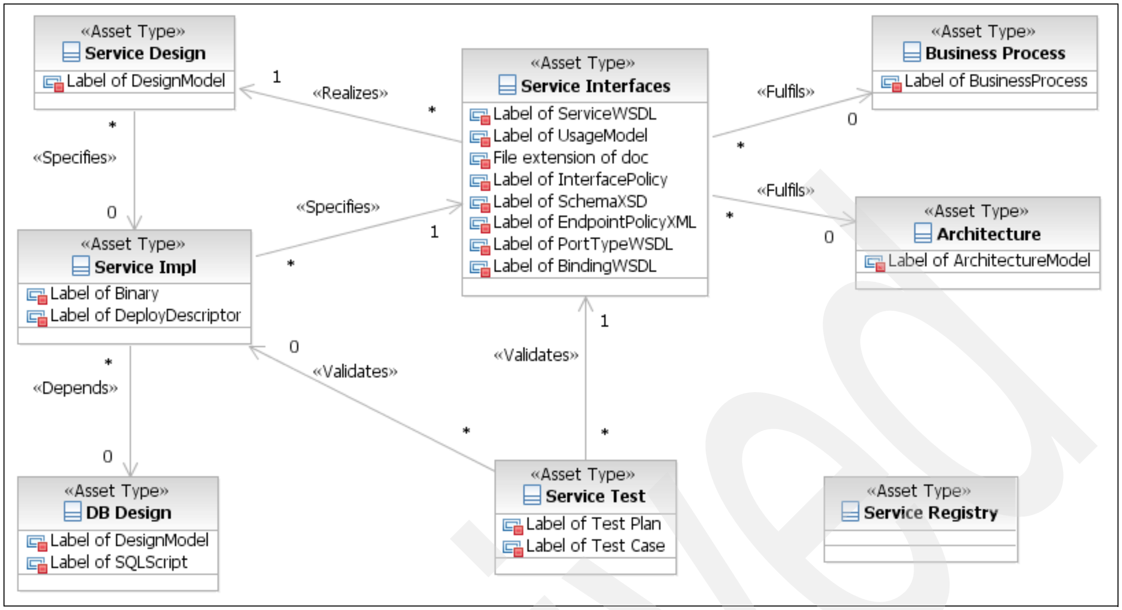


Figure 6-1 Relationships between the asset types related to services

Relationships are defined in Rational Asset Manager for each of the asset types with the artifact constraints shown in the UML diagram as attributes. The attributes indicate what is required to be associated with the asset type to ensure compliance. For example, the *Business Process* asset type requires a file or artifact to be associated with it that has a label of *ProcessModel*.

Additional information about the cardinality of each of the artifact constraints can be found in Rational Asset Manager via the administrative interface.

6.2 Establishing governance in an asset repository

An organization needs to have a Asset Governance Board to identify and plan the asset management system. Assets are grouped by business need and asset type, but they are contained in and shared across communities.

Governance is the establishment of:

- Chains of responsibility to empower people
- Measurement to gauge effectiveness
- Policies to guide the enterprise to meet its goals
- Control mechanisms to ensure compliance
- Communication to keep all required parties informed

The responsibilities for providing governance for asset management activities can be categorized according to the definition of governance in the following lists.

Asset Governance Board responsibilities for chains of responsibilities include:

- Defining asset review workflows
- Defining asset tracking workflows

Asset Governance Board responsibilities for measurement include:

- Defining metrics to track

Asset Governance Board responsibilities for policies include:

- Defining asset rules and policies
- Defining asset types

Asset Governance Board responsibilities for control mechanisms include:

- Participating in Review Boards and asset reviews

Asset Governance Board responsibilities for communication include:

- Preparing reports

Community administrators in Rational Asset Manager (RAM) configure communities, authorize user accounts, set access controls to existing users, define the review process, map forums to change management systems (for example, Rational ClearQuest), and create connections to Rational ClearQuest or WebSphere Service Registry and Repository (WSRR).

All the tasks mentioned are based on each respective community (see Figure 6-2 on page 127).

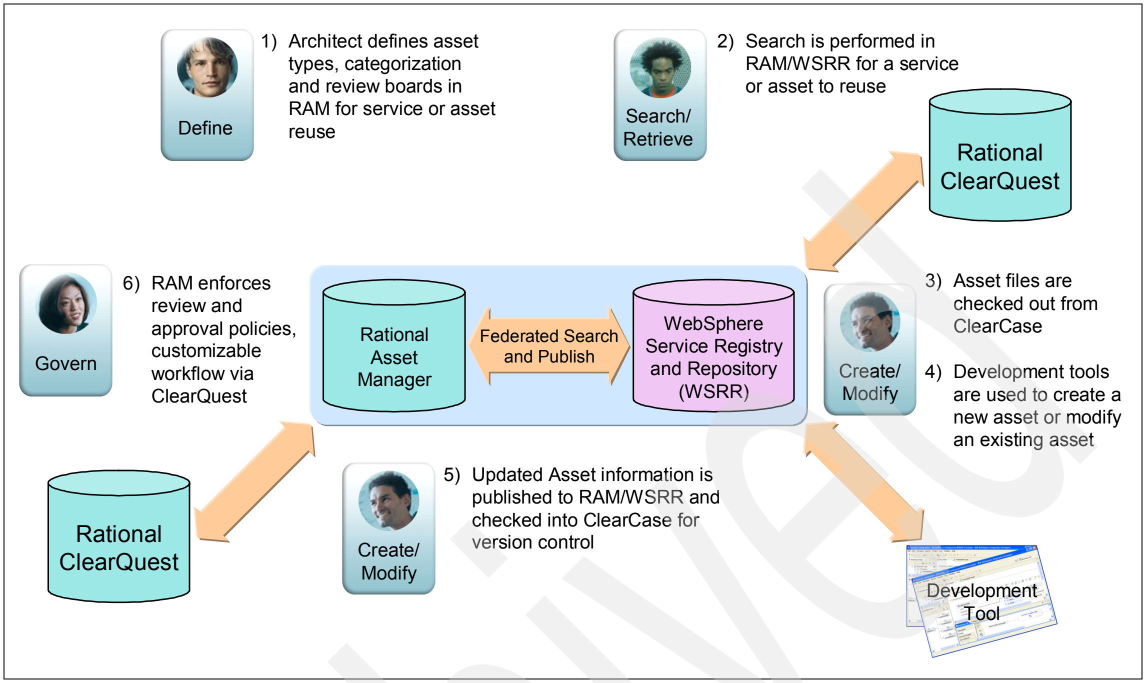


Figure 6-2 Role-based model diagram of Rational Asset Manager and the integration of other tools

The ITSO-Tel project team to implement the workforce management scenario has identified key resources for the governance of the solution in Rational Asset Manager. The key members are shown in Table 6-3.

Table 6-3 Key members of the workforce management team

Member	Role	Responsibility
John	Lead IT Architect	► Administer workforce community ► Reviewer of all assets
Martha	Business IT Architect	► Produce assets for workforce solution ► Consume assets for workforce solution ► Reviewer of Service Identification and Specification assets
Martin	Technical Architect	► Produce assets for workforce solution ► Consume assets for workforce solution ► Reviewer of Service Identification and Specification assets
Ronnie	Program Manager	► Reviewer of Realization assets

Member	Role	Responsibility
Joe	Technical Team Lead	► Informed about the addition of new assets

6.2.1 Creating communities

A *community* is a repository structure that facilitates user interaction with a group of assets that are related in some way. Communities consist of users who can be grouped to facilitate review processes and forums within the community. Community administrators assign roles and permissions to users and user groups, create and manage review processes, and manage request tracking.

Based on their roles and permissions, users can participate in reviews. They can also browse, download, modify, and submit assets.

To create ITSO-Tel's community for support of the workforce management work, log on to the Rational Asset Manager Web Application as an Administrator and perform these tasks:

1. Click the Administration tab.
2. Click **Communities** that appears on the Repository Administration sidebar.
3. Click **New Community** (see Figure 6-3 on page 129):
   a. In the Name text box, enter a name for the community. Here, we have used `Workforce Management` for an example.
   b. In the Description text box, enter a short description, such as `Workspace Management Community` for an example.
   c. Assign a community administrator to this community. Search for a user to be the community administrator and select a name from the list that appears. We selected `john`, the assigned administrator (See Table 6-3 on page 127), assuming the user name has already been created.

   > **Note:** For more information about users, refer to 6.3, "Managing users and groups" on page 129.

   d. Click **OK** to finish.

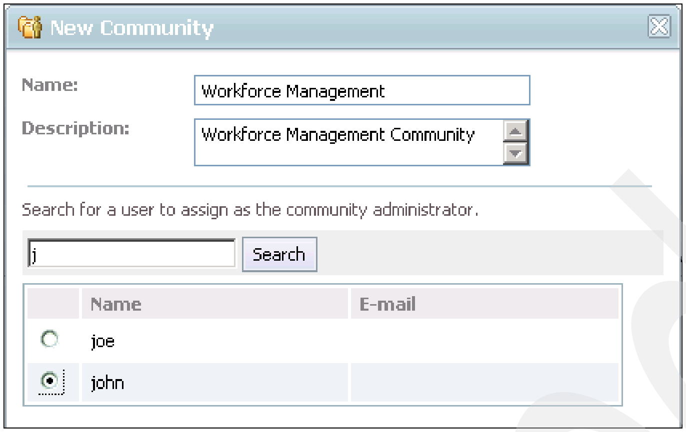


Figure 6-3 Defining a new community

4. To view the details of the community that you have created, click the community name **Workforce Management** from the communities page (see Figure 6-4).

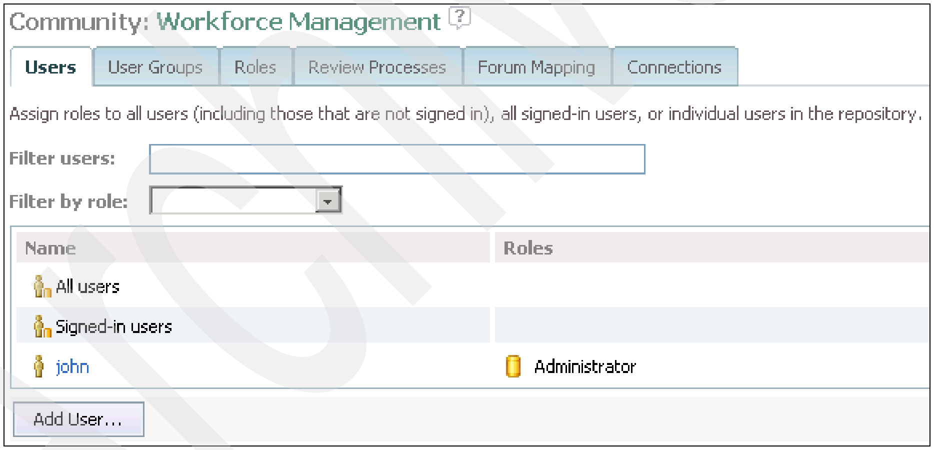


Figure 6-4 Community details

6.3 Managing users and groups

We need to add users and user groups with relevant roles and permissions, who will have access to the assets in the community. From Table 6-3 on page 127, Martha, Martin, and Ronnie need to be added as users to ITSO-Tel's workforce community.

Users

To add users, follow these steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration tab, click the Community `Workforce Management` where you want to add users.
3. Click **Add Users** in the Users tab:

   a. In the Search text box, enter the name or part of the name of the user you want to add in this community and click **Search** (see Figure 6-5). Martha was selected as the first user to be added.

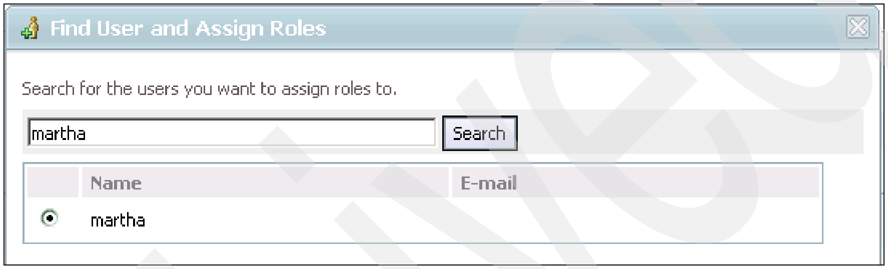


Figure 6-5 Search for the user to add into the community

   b. Select the radio button that appears beside the user name and click **Next**. Here, we have selected `martha`.

   c. Select the roles that you want to assign to the user and click **Finish** to add the user in the community (see Figure 6-6).

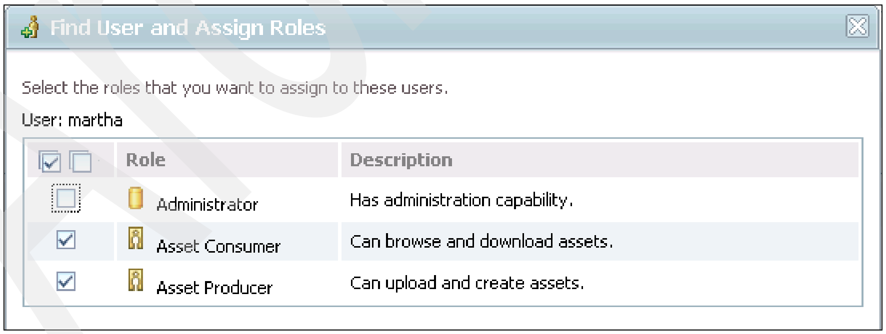


Figure 6-6 Assigning roles for users

4. Repeat Step 3 to add Martin and Ronnie to the user community.

User groups

In a community, *user groups* are a collection of users who share the same roles and permissions. For review processes, user groups can be added as a group of reviewers.

ITSO-Tel's architects are required to be identified and added to the SOA architecture group defined by their organizational structure. Currently, three key members have been identified (See Table 6-3 on page 127).

To add user groups, follow these steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration tab, click the community `Workforce Management` where you want to add the user group.
3. On the User Groups tab, click **New User Group**:
   a. In the Name text box, enter the name of the user group. We have used `SOA Architecture Group` here as an example.
   b. In the Description box, which is optional, you can enter a few words about the purpose of the group you are creating. We have used this description as an example:
      ```
      Plan, model, review SOA services through identification,
      specification, realization etc
      ```
   c. To associate users in the group, click **Add Users** (see Figure 6-7 on page 132).

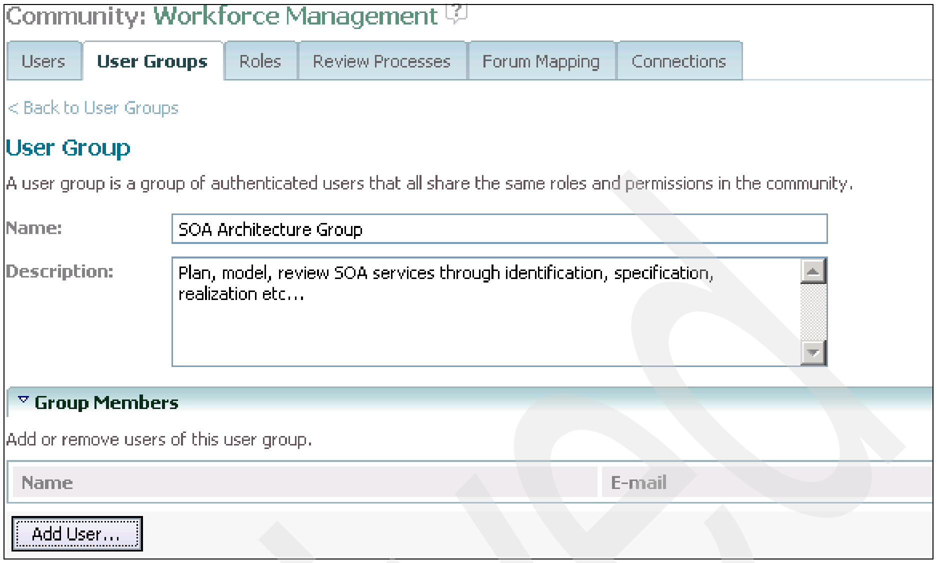


Figure 6-7 Defining a user group in a community

4. In the Search text box, enter the name or part of the name of the user that you want to add in this user group and click **Search**. Here, we have used `m` to display `martha` and `martin` as an example:
   a. Select the check box that appears beside the user name and click **Add**. Here, we have selected both `martha` and `martin` for examples (see Figure 6-8).

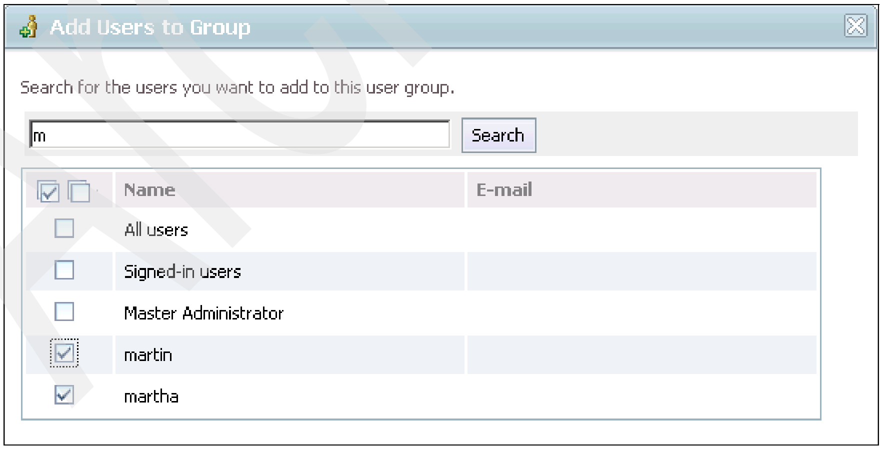


Figure 6-8 Adding users to the user group

5. Select the roles for this group from the Group Roles section (Figure 6-9). Here, we have selected `Asset Producer` and `Asset Consumer` as examples. Click **OK** to finish. If the roles are not present, you can go to the Roles tab to create new roles for the community (see 6.3.1, “Creating roles for a community” on page 133).

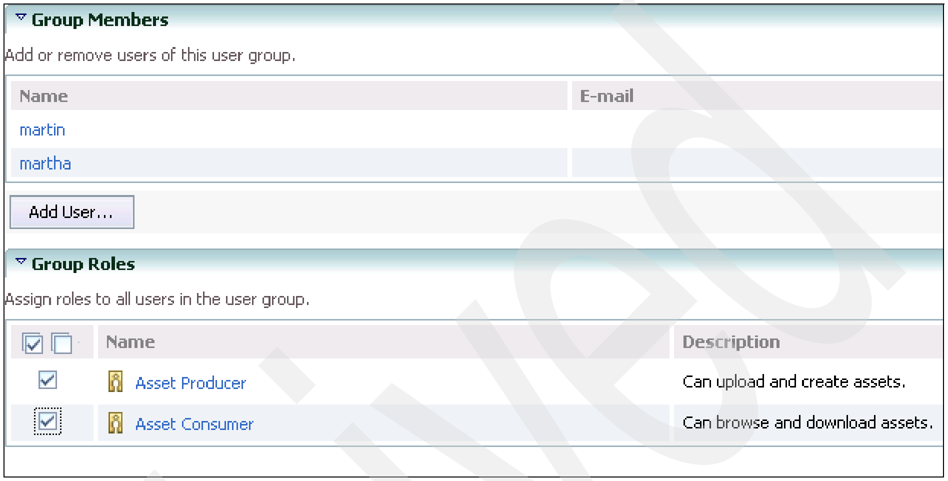


Figure 6-9 Assigning roles to the user group

6. Repeat step a on page 132 to add John as part of this user group.

6.3.1 Creating roles for a community

Roles are a set of specific permissions, which can be imposed on a single user or a user group in the community.

ITSO-Tel requires a role for performing the review of all SOA strategic assets to ensure compliance in delivery of the solution.

To define ITSO-Tel’s strategic review role, follow these steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration page, select the community `Workforce Management` and click the **Roles** tab to display the list of roles.
3. Click **New Role** and the window will refresh with the following options (Figure 6-10 on page 134):
   a. In the Name text box, enter a name for the role you are creating. Here, we have used `SOA Strategy Reviewer` as the name.

b. In the Description text box, you can briefly describe the role: `User can search, read, browse, Subscribe and download SOA strategy assets for reviewing.`

c. In the Role Permissions section, select the appropriate permissions that you want this role to have. ITSO-Tel requires the reviewer to have `Browse assets`, `Download assets`, `Read asset details`, `Search assets`, and `Subscribe to assets` for reviewing.

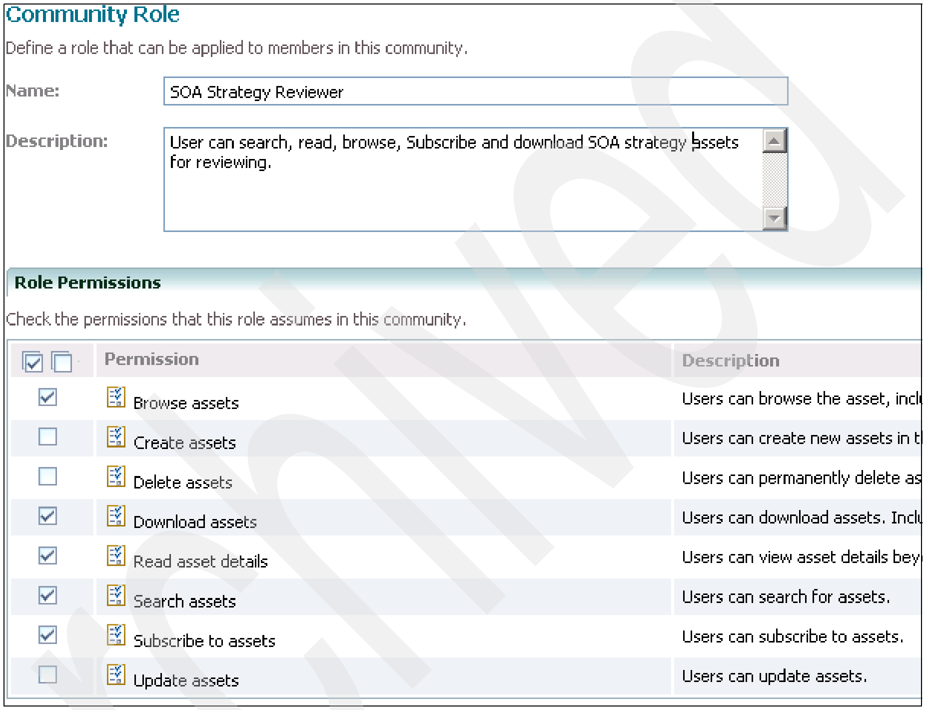


Figure 6-10 Defining a role in a community

4. In addition to this, you can also add one or multiple conditions to define the scope of the role. You can categorize a role by Asset Types, Asset Categories, or both, in more than one combination. This is also called a *constraint* to the role. Creating multiple constraints for roles gives the repository advanced levels of governance:

   a. In the Role Scopes section (Figure 6-11 on page 135), use the lists to create the conditions for the constraints.

   b. Click **Add Constraint** to include them in the role. You can create as many constraints as needed to define the permissions in a role. Creating multiple constraints for roles gives the repository advanced levels of governance. Here for example, we have used three constraints, such as

`Asset category must be ITSO-Tel Business Domain, Asset Type must be Service Interface,` and `Asset Type must not be test.`

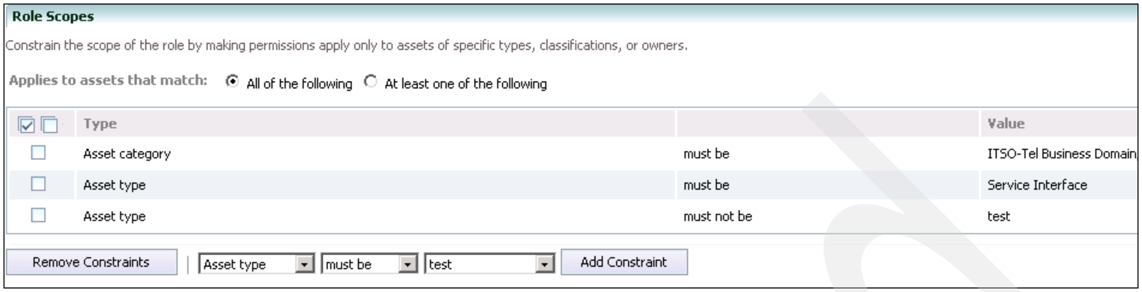


Figure 6-11 Applying constraints in a role

5. Click **OK** to finish creating the role.

6.3.2 Defining the review process and review board

The *review process* provides governance over assets or services in the asset repository.

It is a framework for reviewing assets where you can approve, retire, or delete assets. Users can review and verify the quality and accuracy of assets according to their specific areas of expertise.

You can create many different levels of governance by using review processes. You can create review processes primarily by asset type, but you can also constrain the review process to apply to specific asset types in specific categories.

After determining the control mechanisms to govern what is needed in the repository, you can plan to conduct many reviews before the repository is available for its target consumers. In this task, ITSO-Tel has defined the SOA Architecture group, the Business Analysts, the SOA Program Managers, and Practitioners who are required to participate in one or more review workflows (see Figure 6-12 on page 136).

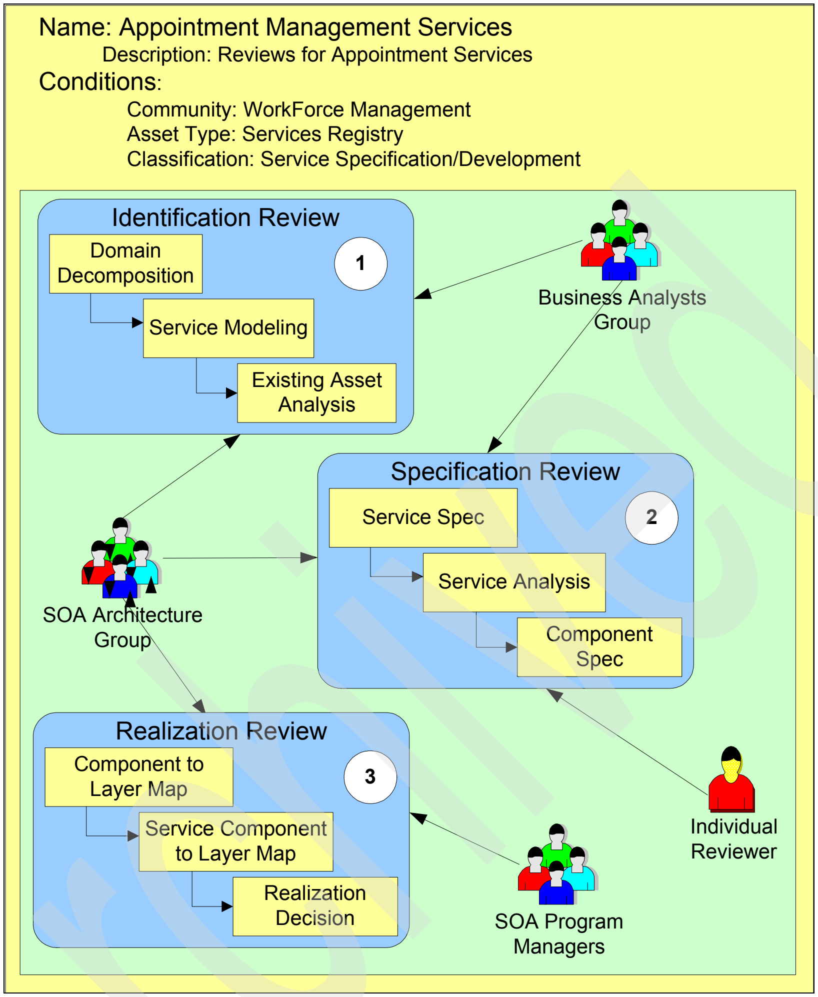


Figure 6-12 Example of review processes for an asset type

As shown in Figure 6-12, the review workflow can be conducted in order of relevance to an asset. Here, the identification review workflow is conducted first, followed by two other reviews. It is not required to follow all the reviews every time, because the reviews will vary based on the type of assets. The types of reviewers for each of the reviews will likely vary, because special skills might be required. You must define the conditions for selecting a review workflow.

To create ITSO-Tel's review processes using either the default review process or a process driven by ClearQuest, follow these steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration page, select the community `Workforce Management` and click **Review Process** tab.
3. Click **New Review Process**:
   a. To create a default review process type, select **Default review process** from the list and click **OK** (see Figure 6-13):

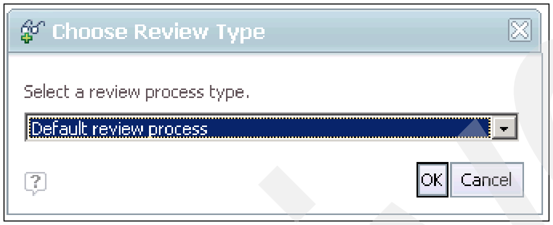


Figure 6-13 Select default review process type

   b. In the Name and Description text boxes, you can specify the name of the review process and a short description for it. Here, we have used `Identification Review` as the name of the review process and `This process will identify whether the goal and components of the service have been identified properly` as our description (see Figure 6-14).

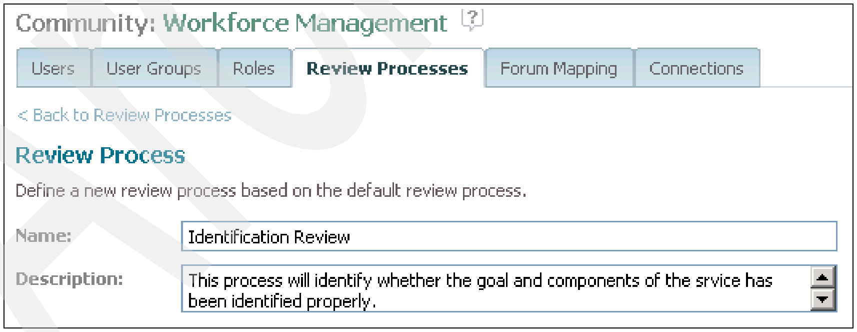


Figure 6-14 Naming and describing the review process

4. Next, in the Conditions section, you can specify the conditions for which you want to use the review process. You can select as many conditions as you want to increase the level of governance and accuracy. To do this, select the

Asset type or optionally Categories from the list and click **Add** to include the condition in the review process (see Figure 6-15):

> **Note:** After clicking Add, wait for the window to refresh for Asset type and then also for Category.

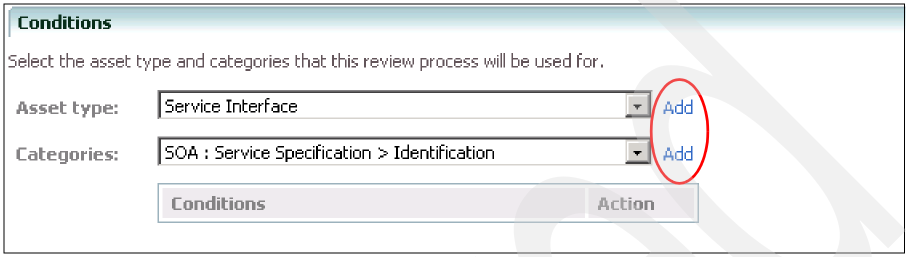


Figure 6-15 Specifying the conditions of a review process

> **Note:** The review process can only be applied to a single *Asset type,* but multiple *Categories* can be included.

a. The next task is to add review board members who will provide the final approval that is needed to push the asset from the review state to the approved state. If a review board member rejects the asset, it will go back into the draft state and require revisions. In the Customization section, click **Add Review Board Member** and type a user's name to add as the review board member. You can add as many review board members as needed for this type of review (see Figure 6-16 on page 139).

b. Following the same steps, you can also add individual reviewers by clicking **Add Reviewers** (see Figure 6-16 on page 139). From Table 6-3 on page 127, we allocated Martin, Martha, and John to this process.

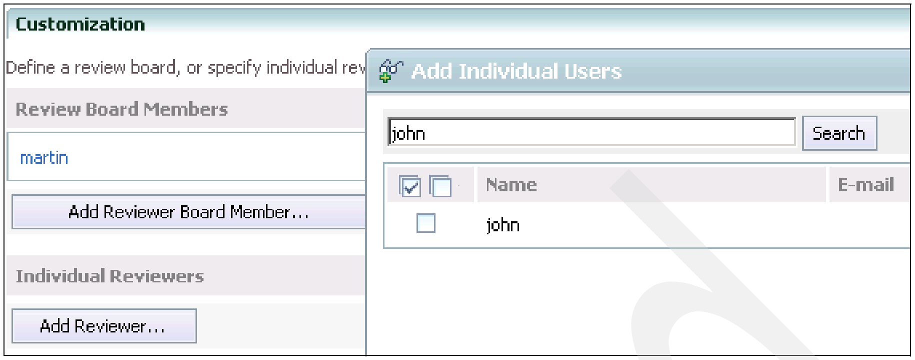


Figure 6-16 Adding review board members and individual reviewers

5. Repeat step b on page 138 to add Martha as a reviewer board member:

   a. Optionally, if you have created a user group for the review process, you can click **Add User Group** to add a user group for reviewing the assets (see Figure 6-17). In ITSO-Tel's case, the Business Analysts can also perform reviews at the service identification. Moreover, you can select the **Make reviews private** check box so that reviewer names are not visible to other users. Again, you can also set reminders to send e-mail notification on the schedule that you choose in the **Reminders** list. For example, `1 Days` will send an e-mail reminder every day to the reviewers that you added in the review process.

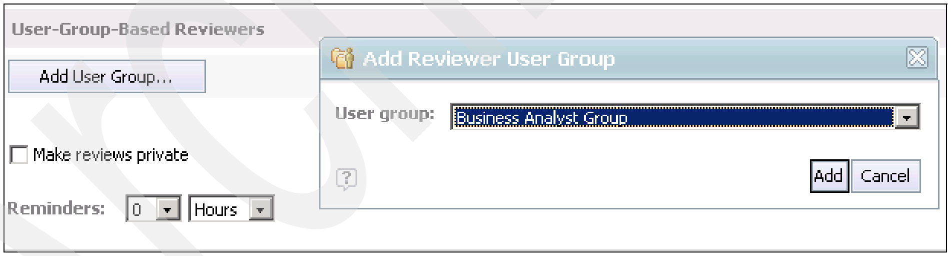


Figure 6-17 Adding review user group

6. Click **OK** to finish adding the default review process:

   a. For creating a ClearQuest driven review process, select **ClearQuest driven review process** from the list and click **OK** (see Figure 6-18 on page 140).

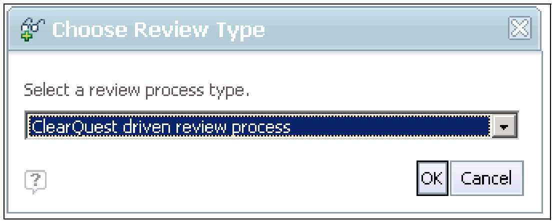


Figure 6-18 Select ClearQuest driven review process type

b. Follow the exact steps previously discussed in the Default review process for naming, condition, and customization sections. In this case, we have an additional section called Mapping, which is described next:

i. For mapping a ClearQuest workflow as a review process, in the Mapping section, click **Change** in the Connection field.

> **Note:** We assume that you have already defined a ClearQuest recordtype to match the review process and also that you have established connections with ClearQuest from the connections tab (see 6.3.4, “Linking communities to a Change Management System and WebSphere Service Registry and Repository” on page 147).

ii. Select a connection from the list that you want to use and click **OK** (see Figure 6-19).

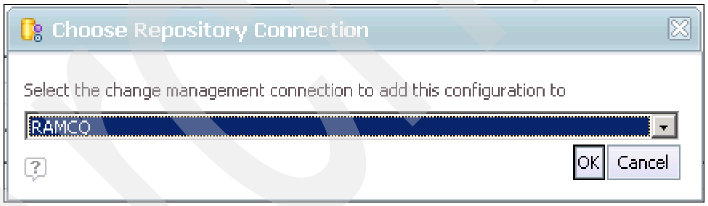


Figure 6-19 Selecting ClearQuest connection for review process

7. In the Record type field (Figure 6-20 on page 141), select the record type where the review process workflow is defined. Wait for the window to be updated:

   a. In the Approved and Rejected field list boxes, select the end states as defined in the ClearQuest record type.

   b. In the Comments field, select a field name which will store the review comments.

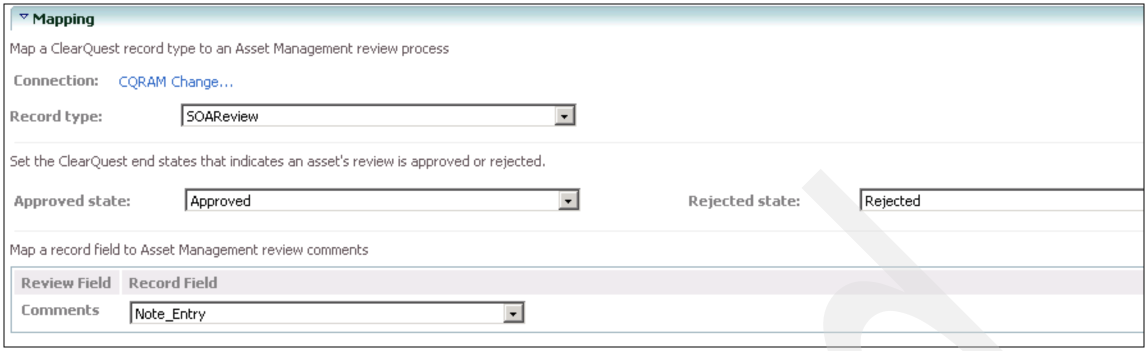


Figure 6-20 Mapping ClearQuest workflow to a review process

c. You can add other mandatory or optional fields that are required to have some values from Rational Asset Manager during the submission of a review, which will be used by reviewers. Click **Update Fields** and click **OK** after completing the selection by using the side arrows in the middle of the Add Field dialog box (see Figure 6-21).

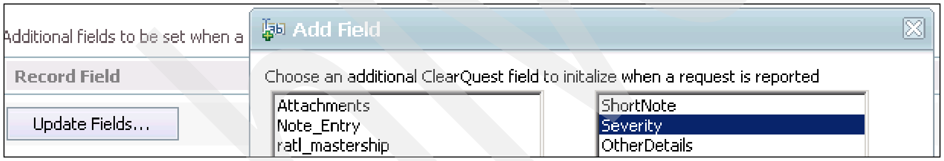


Figure 6-21 Choose additional fields for ClearQuest driven review process

8. To specify the values with which those fields will be populated, click **Insert value** to select values from a list, type static information in the text box, or both. You can also use Rational Asset Manager system variables to reflect in the ClearQuest record (see Figure 6-22 on page 142).

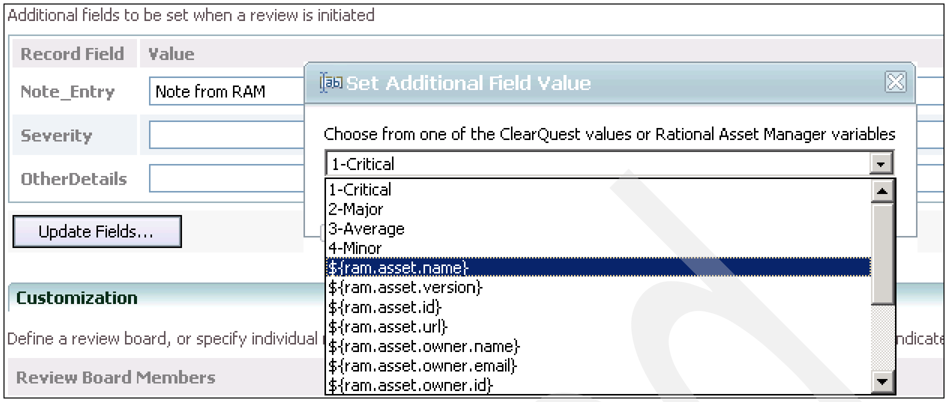


Figure 6-22 Specifying values for ClearQuest fields in the review process

9. Click **OK** to finish adding the ClearQuest driven review process.

6.3.3 Managing forums

Forums are essentially used to discuss assets with other users of the repositories. A discussion topic is a high-level topic of discussion and contains discussion posts.

Repository administrators and asset owners define the types of forum discussions for each asset. Asset reviewers, consumers, browsers, and whoever has access can provide feedback in the discussion forums. The forum topics can vary from very general discussions, change management, and defect tracking to implementation questions, to executive strategy.

ITSO-Tel requires a forum for the discussion of defect tracking associated with assets.

To configure or adding ITSO-Tel's defect tracking forum, follow the steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration page, select the community `Workforce Management` and click the **Forum Mapping** tab.
3. Click **New Configuration** and wait for the window to refresh.

Note: We assume that you have already defined a ClearQuest recordtype to match the discussion topic and also that you have established connections with ClearQuest from the connections tab (see 6.3.4, “Linking communities to a Change Management System and WebSphere Service Registry and Repository” on page 147).

4. In the Choose Repository Connection window, select the connection name you want to use and click **OK** to proceed (see Figure 6-23):

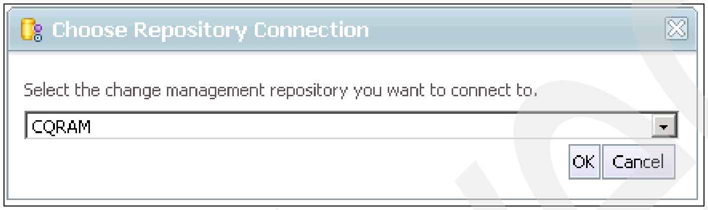


Figure 6-23 Selecting ClearQuest connection for adding forums

   a. In the Name and Description text boxes, you can specify the name of the forum and a short description of it. Here, we have used `Defect Tracking` as the name of the review process and our description is `Defect Management` as an example (see Figure 6-24).

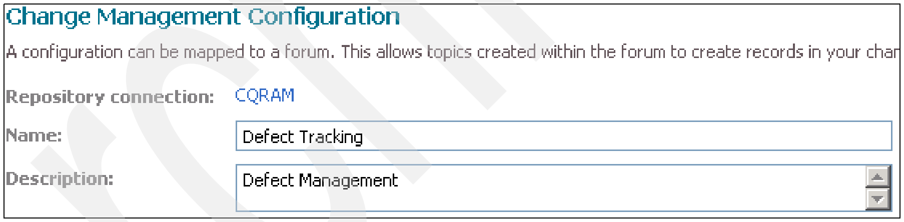


Figure 6-24 Naming the forum

5. In the Mapping section in the Record type field, select the ClearQuest record type that is appropriate for this forum. We have selected the **Defect** record type as an example here:

   a. In the “Map record fields to request fields” section, you can specify the fields from the ClearQuest record that will be filled in using information from the forum topic and response posts (see Figure 6-25 on page 144).

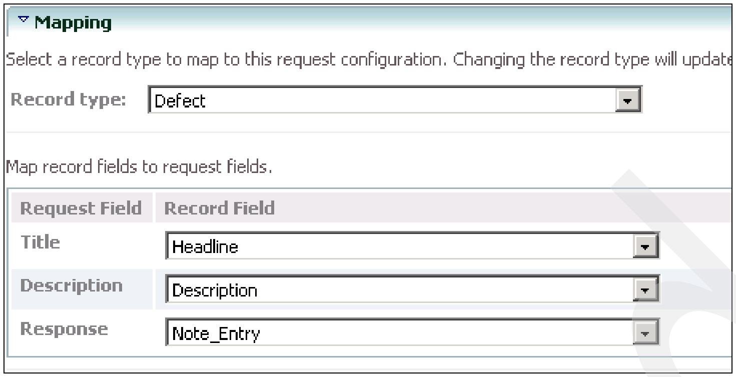


Figure 6-25 Specifying fields to map record fields with request fields

6. In the additional record fields section, you can optionally specify additional fields in the ClearQuest record that will be automatically populated with data from the asset or the forum. You can include static text, variables, or both. Clicking the Insert Value link will retrieve suggested values from ClearQuest along with common variables from Rational Asset Manager. Here, we have used `3-Average` for the Severity field for an example (see Figure 6-26).

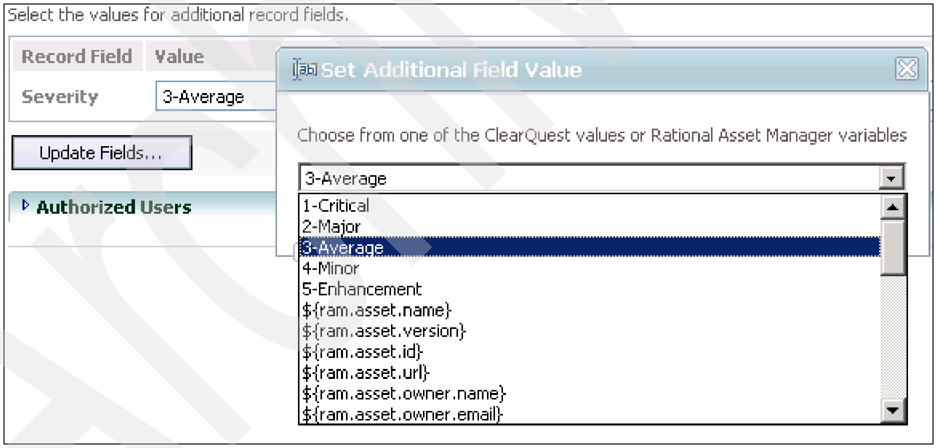


Figure 6-26 Specifying additional fields to map record fields with request fields

7. Click **OK** to specify the field value.
8. Click **OK** to complete defining the forum.
9. You can also add queries of ClearQuest records from the Rational Asset Manager Web application.

> **Note:** You must have the community administrator role to add a new connection, and there must be a forum mapping configuration for the community.

10. Click **New Query** and select the forum name from the “Choose Changes Configuration” dialog box and click **OK** to display the query definition page (see Figure 6-27).

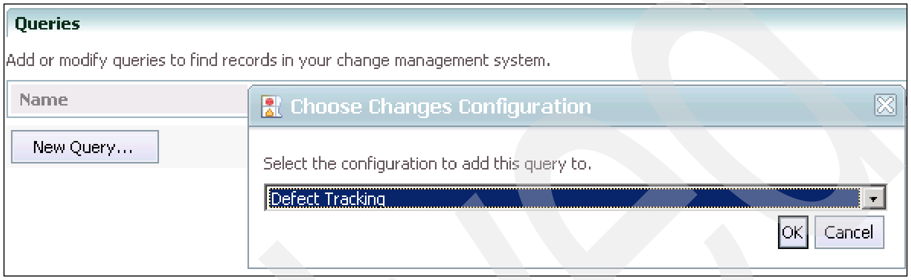


Figure 6-27 Select forum configuration for queries

11. In the Name and Description text boxes on Figure 6-28 on page 146, you can specify the query name and a description. Here, we have used `All Defects` as an example.
12. Click **Select Query** to load queries from the ClearQuest database. In the “Choose Query” dialog box, navigate via the Public Queries to the query that you want to select and click **OK** to add it.

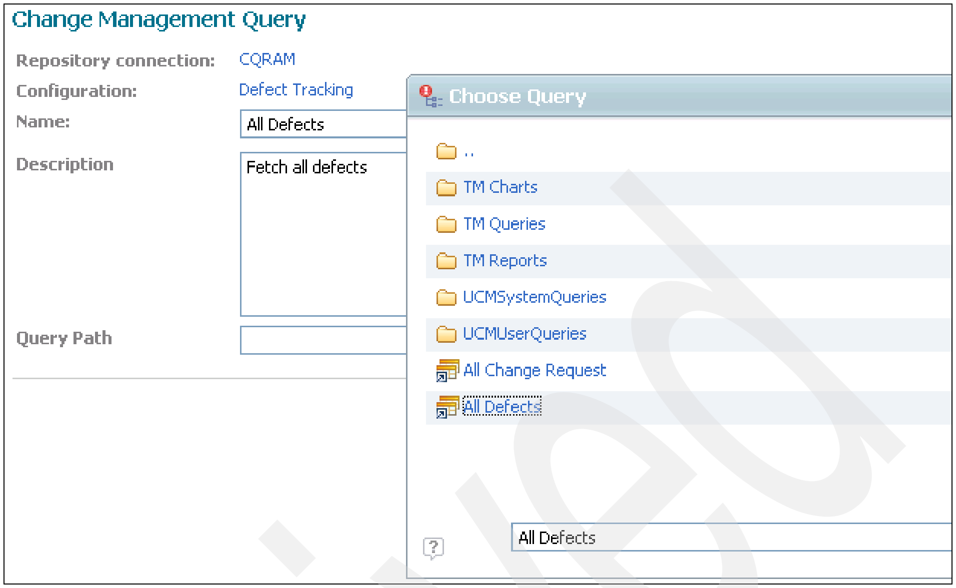


Figure 6-28 Selecting a query

13. Click **OK** to finish configuring the query.

14. After you finish adding the forum and the query in the Forum Mapping tab, you will be able to see the forum and query links (see Figure 6-29).

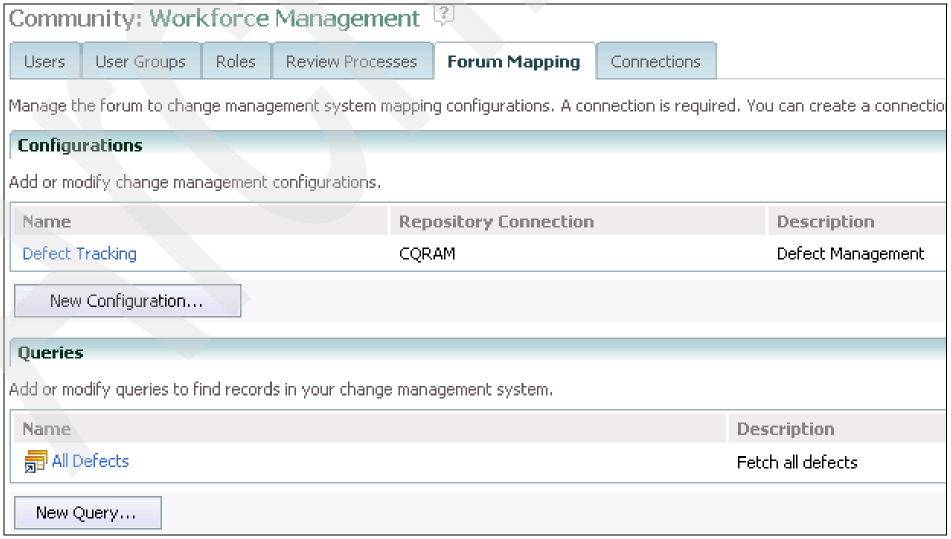


Figure 6-29 The forum mappings and selected queries

6.3.4 Linking communities to a Change Management System and WebSphere Service Registry and Repository

A *repository* can integrate with various tools, such as SCM tools, change management tools, service registries, and so on. These tools can be integrated with the repository to provide a seamless interaction for the service development workflows. This also provides a governed asset management environment to preserve and improve the organization's software or service-related investments.

You can create connections to Rational ClearQuest through the Rational Asset Manager Web application. Connections are essentially required for forum mappings, queries, and ClearQuest-driven review processes.

You can create a connection to WebSphere Service Registry and Repository using the Rational Asset Manager Web application for publishing Web services assets.

ITSO-Tel has an existing Rational ClearQuest installation that governs the change management process in the organization and requires this existing Rational ClearQuest installation to be integrated with Rational Asset Manager. In addition, ITSO-Tel wants to integrate the deployment of assets to a deployed WebSphere Service Registry and Repository operating in a production environment.

Creating connections to ClearQuest

A *schema* in Rational ClearQuest is a complete description of the process model for one type of change tracking (the metadata for that process model). It includes a description of the life cycle of the process, the structure of the data, hook code, user interfaces, reports, and queries.

Rational Asset Manager integrates with Rational ClearQuest so that all members of an organization have the opportunity to provide feedback on assets at any time during an asset's life cycle (Figure 6-30 on page 148).

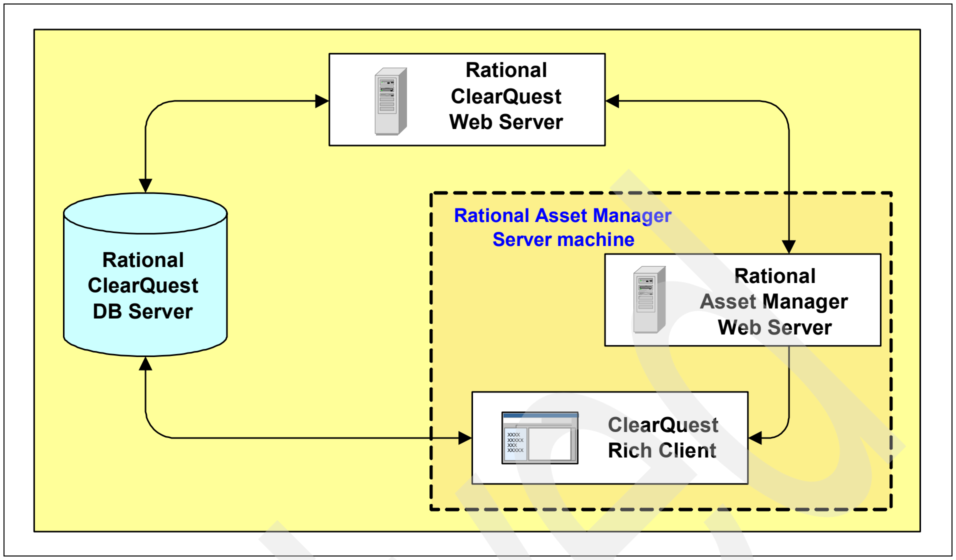


Figure 6-30 ClearQuest and Rational Asset Manager architecture

To establish a ClearQuest connection to an existing ITSO-Tel installation, follow these steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration page, select the community `Workforce Management` and navigate to the Connections tab.
3. Click **New Connection** on the Change Management Connections section.
4. A Choose Repository Type dialog box will appear, select **ClearQuest** and click **OK** (see Figure 6-31):

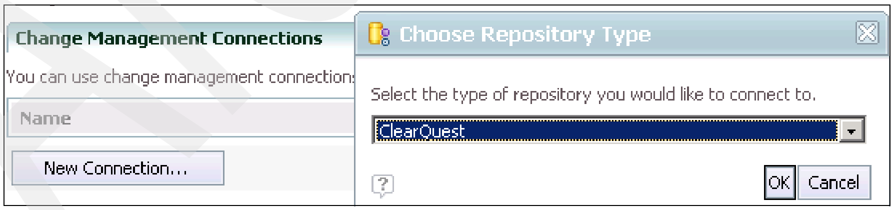


Figure 6-31 Creating ClearQuest connection

   a. In the Repository Connection Details window, provide (see Figure 6-32 on page 149):

      i. Name and description.

ii. Schema repository: This is the database that is used to store and manage a group of schemas, including all versions of those schemas and their associated user databases. This list is populated from the list of connections set up in the Rational ClearQuest Maintenance Tool on the server machine.

> **Note:** The name used for the Schema repository should be the same name configured on the ClearQuest server. If you use a different name, you will not be able to access the information from the Rational Asset Manager server on the ClearQuest server.

iii. Database: This is the particular user database within the schema repository to which you want to connect.

iv. CQWeb URL: This is the URL to the Rational ClearQuest Web application.

v. User name: This is the user ID of a user who is authorized to access Rational ClearQuest. This user will own any requests opened in the external repository.

vi. Password: The user’s password for accessing Rational ClearQuest.

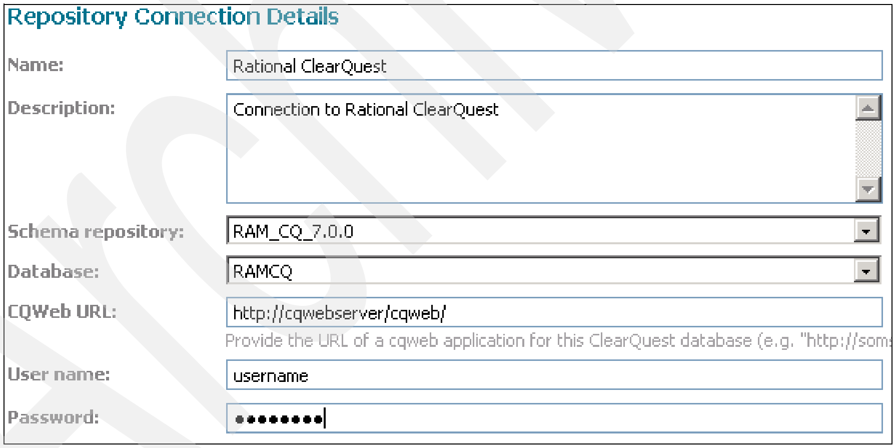


Figure 6-32 ClearQuest connection details

5. Click **OK** to complete.

Creating a connection to WebSphere Service Registry and Repository

WebSphere Service Registry and Repository is the master metadata repository for runtime service descriptions.

As the integration point for service metadata, WebSphere Service Registry and Repository establishes a central point for finding and managing service metadata that is acquired from a number of sources, including service application deployments and other service metadata, and endpoint registries and repositories.

After that happens, visibility is controlled, versions are managed, proposed changes are analyzed and communicated, usage is monitored, and other parts of the SOA Foundation can access service metadata.

Rational Asset Manager manages information that is useful for developing, reusing, and managing services as assets. It provides the ability to synchronize category schemas with WebSphere Service Registry and Repository classification systems for proper categorization of services, provide traceability and details, and includes collaboration software.

WebSphere Service Registry and Repository manages information that is useful for runtime operation, management, and the development use of services. It gives you the ability to select service endpoints dynamically in an SOA run time, govern runtime changes to service metadata, set and get runtime policies for service execution, and get deployed service details, such as endpoints and service definitions. Together these products create seamless, reusable asset tracking mechanisms through development and runtime activities.

To establish a WebSphere Service Registry and Repository connection to ITSO-Tel's production repository, follow these steps:

> **Note:** Information about publishing to WebSphere Service Registry and Repository is provided in 6.8, "Service publication" on page 180.

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration page, select the community `Workforce Management` and click the **Connections** tab.
3. Click **New Connection** on the Web Service Registry Connections section.

4. On the Connection Properties page, complete the form with the details for your WebSphere Service Registry and Repository installation (see Figure 6-33 on page 152):
   a. Name and description
   b. Host: The fully qualified name of the WebSphere Service Registry and Repository server, for example: server.yourcompany.com. Do not include http:// or https://. To use a secure connection, check Secure Mode.
   c. Port: The port number to use, for example, `9080`.
   d. Login: This is the user ID of any user who is authorized to access WebSphere Service Registry and Repository.
   e. Password: The user's password for accessing WebSphere Service Registry and Repository.
   f. Test Connection: After you have entered the host, port, user name, and password, you can click **Test Connection** to check whether a connection can be established with the server.
   g. Default Asset owner: Select the user who will own assets that are automatically added to the Rational Asset Management repository during synchronization with WebSphere Service Registry and Repository. The default asset owner will be assigned to an asset when the user ID in WebSphere Service Registry and Repository is not found in Rational Asset Manager and when a WebSphere Service Registry and Repository user ID cannot be created. When you select the default asset owner, verify that the user has the permission to create an asset in Rational Asset Manager.
   h. Indexing interval: Type the time interval for the synchronization to start.

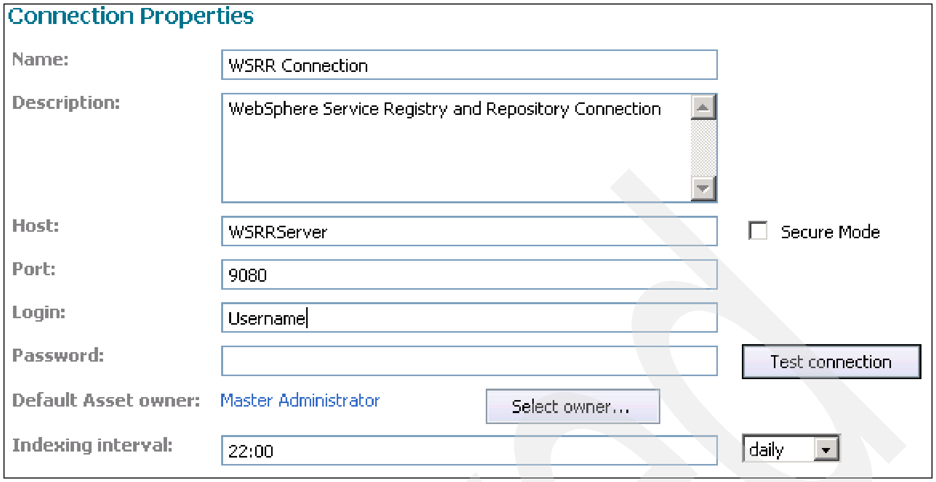


Figure 6-33 WebSphere Service Registry and Repository connection details

5. Click **OK** to finish.
6. To synchronize WebSphere Service Registry and Repository with Rational Asset Manager, click **Synchronize** on the same page (see Figure 6-34).

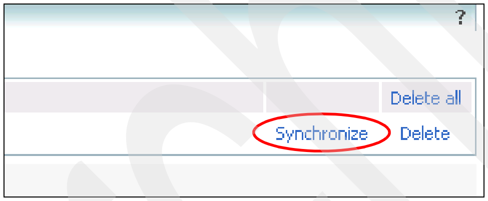


Figure 6-34 Synchronize link in the Web Service Registry Connections section

6.4 Asset types

When an service or component is submitted as an asset, the asset type must be specified, which will dictate the type of artifacts that are expected for the asset.

Each asset is of a single type and it describes the purpose of the asset, the artifacts that it contains, relationship with other assets, category schemas, context and custom attributes, or metadata elements.

The asset type defines the expected structure, content, relationships, and so forth. The asset is an instance of one type, adhering to the constraints specified by the asset type. The model that ITSO-Tel has proposed for the structure of asset types, assets, and artifacts is shown in Figure 6-35 on page 153.

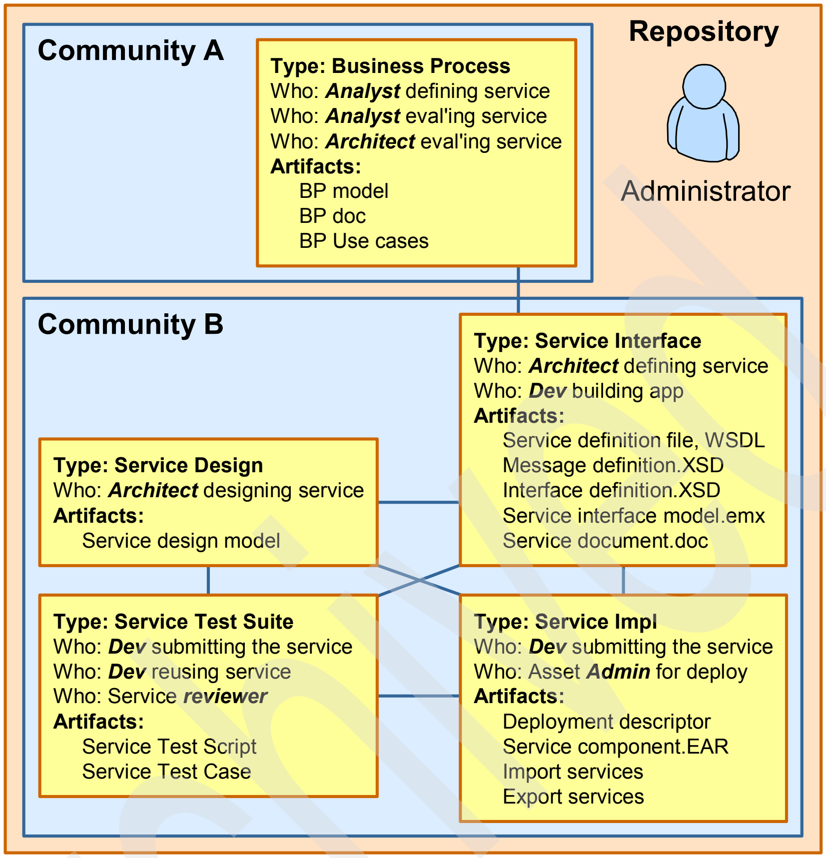


Figure 6-35 Asset type, asset, and artifact relation example model

To create an ITSO-Tel asset type, follow these steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and navigate to the Administration tab.
2. In the Administration tab, click **Asset Types** in the Repository Administration section.
3. Click **New Asset Types**:
   a. Enter a name in the Name field for the Asset Type.
   b. Enter a brief description in the Description field for the Asset Type (see Figure 6-36 on page 154).

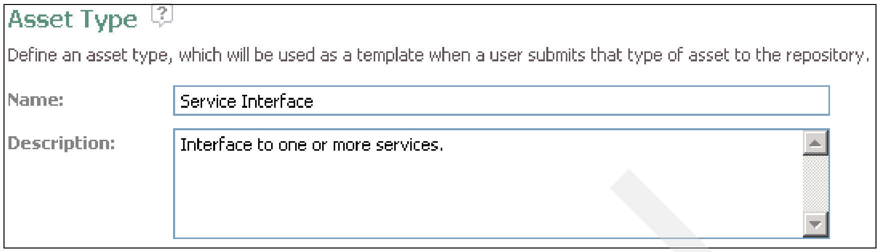


Figure 6-36 Defining an Asset Type

4. You can optionally specify multiple constraints for an asset type, each of which can refer to a specific format or label or extension:

   a. Click **New Artifact Constraint** to display the New Artifact Constraint dialog box (see Figure 6-37).

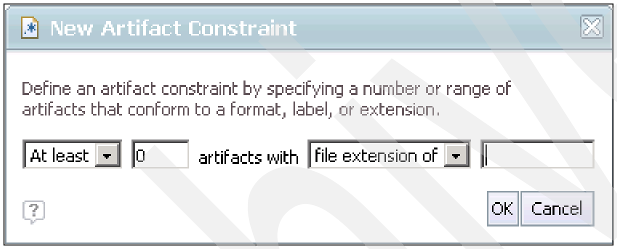


Figure 6-37 Specifying the artifact constraint

5. Select the range of artifacts to be required, choose the restriction type, and then type a value for the type of artifact to require. In Figure 6-38, we have shown some examples.

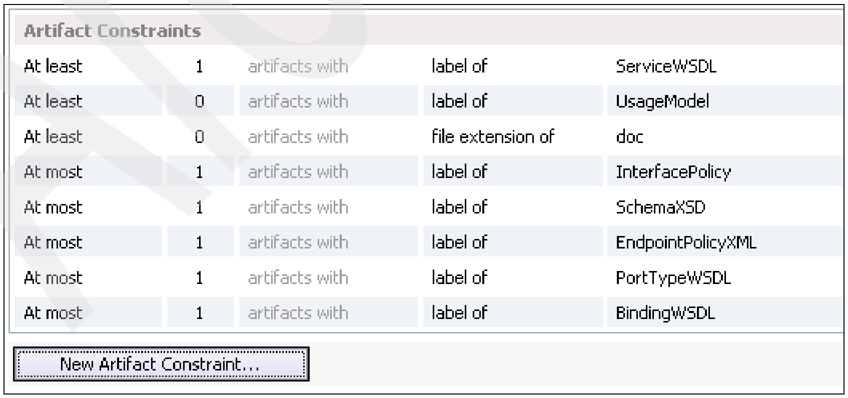


Figure 6-38 Examples of artifact constraints of an asset type

6. You can optionally add category constraints, which can define the relevance of the asset type to a particular category. You can constrain the asset type to apply to either all category schemas or only a subsection of the category schemas that you created in the repository (see Figure 6-39):

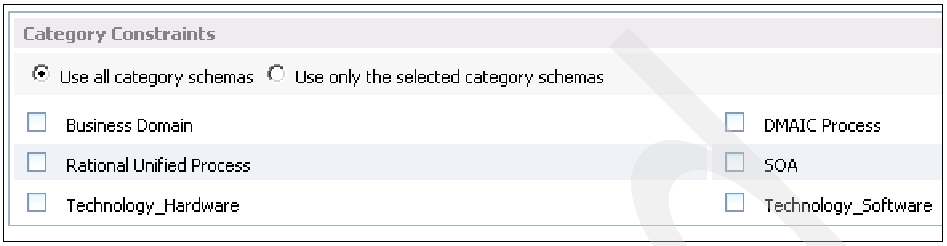


Figure 6-39 Category constraints

   a. To constrain the asset type to all categories, select **Use all category schemas**.

   b. To constrain the asset type to a specific category, select **Use only the selected category schemas** and select from the categories available for this asset type.

7. Optionally, you can also define the relationship constraint which can be relations to other assets, dependencies, other assets, and so forth:

   a. To create a relationship constraint, click **New Relationship Constraint** to display the New Relationship dialog box (see Figure 6-40).

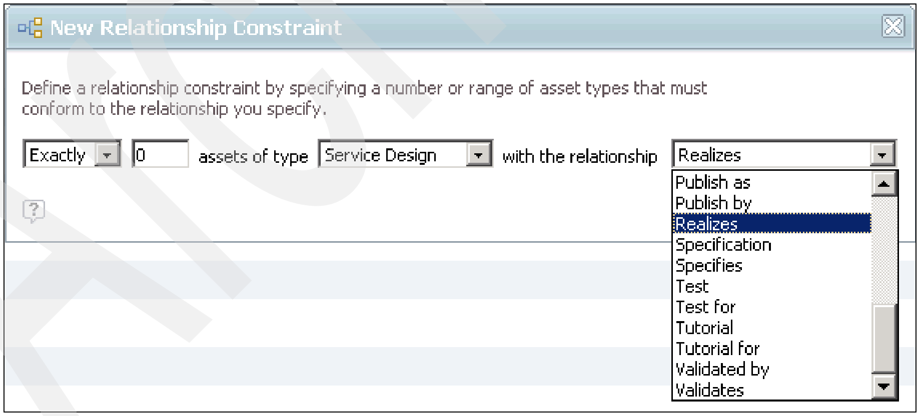


Figure 6-40 Specifying the relationship constraints

8. Specify a quantifiable range and then choose the asset type that must fit or match to the relationship that you choose.

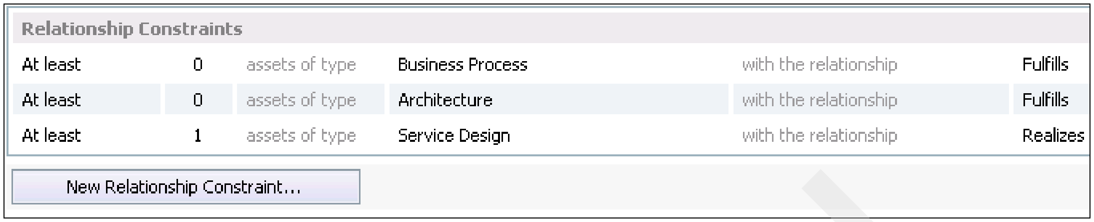


Figure 6-41 Relationship constraints to an asset type

9. Optionally, define an attribute constraint (Figure 10):
   a. Select the check box for an attribute to allow users to define a value for that attribute.
   b. Check the **Required** check box to make this value required for an asset when it is submitted.

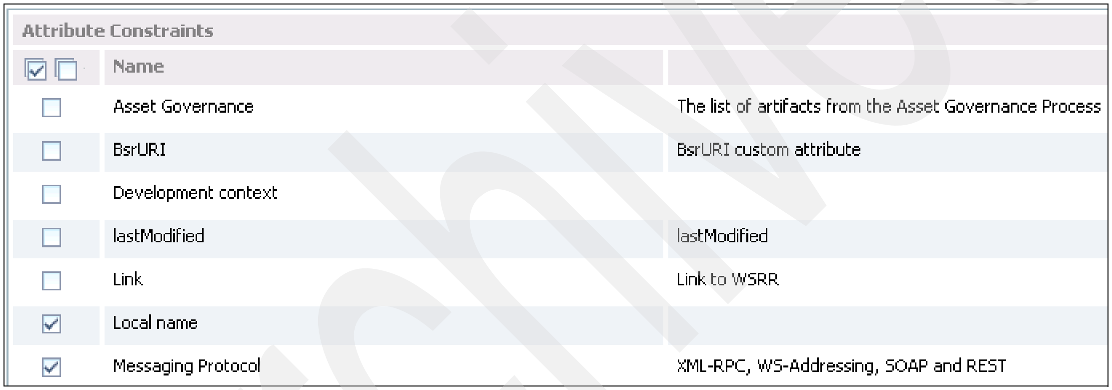


Figure 6-42 Specifying attribute constraints to an asset type

10. Click **OK** to save the asset type.

6.4.1 Category schemas

Category schemas are a way to describe assets. You can use category schemas as mechanisms for searching and discovering assets that have been identified by ITSO-Tel for use in applying their governance approach (Figure 6-43 on page 157). The repository can have many category schemas. The administrator submits these category schemas in the repository.

An asset can be categorized using values from many classification schemas. The category schemas need to be exclusive and cannot be related to each other. Using category schemas in this manner permits many perspectives to be captured. For example, not only do practitioners (architects, developers, testers, and so forth) work with assets, but administrators and technical management will interact with the assets, as well as with report generators, metrics collectors, and other tools.

Each category schema needs to be created with a specific focus. For example, you can create a classification schema to describe the business domain of interest to the enterprise. You might create another category schema to describe the technical contexts for the enterprise, such as runtime platforms and development platforms.

Category schemas are created from the perspective of a target consumer. For example, architects might want to discover an asset's relevancy from the perspective of nonfunctional characteristics.

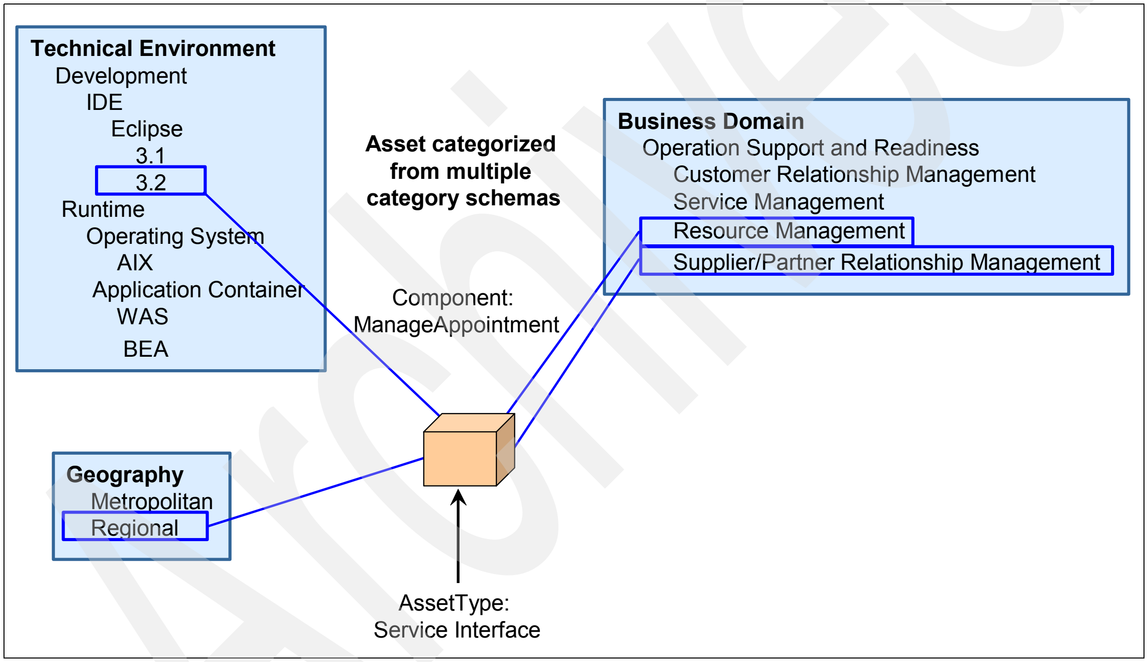


Figure 6-43 Asset categorized from multiple category schemas

There is a similarity and a relationship between asset type and category. Both asset type and category are used for describing assets, and both of them are used for searching and discovery. Categories can be broad or specific. Broad categories can break down into absolute specific nodes, or *children*.

You can create category schemas to organize assets in a hierarchical order within the repository.

To maintain a logical order within a category schema, administrators can designate children as exclusive. When you mark children of a category node exclusive, it means that a user can only categorize an asset using only one of the subcategories, or child categories, rather than more than one of the children. For example, if an asset is found in the transportation schema and is categorized as a car, the subcategories can be red or blue. An asset cannot be both red and blue, so the car category must be marked with exclusive children.

To create a ITSO-Tel's category schema, follow these steps, which must be performed by a user with repository administration permissions:

1. Log on to the Rational Asset Manager Web Application as an Administrator and click the **Administration** tab.
2. In the Administration page, click **Category Schemas** in the Repository Administration sidebar.
3. Click **New Category Schema** to create a category schema.
4. Describe the new category schema (see Figure 6-44):
   a. Enter a unique name for the schema in the Name field. Here, we have used `Business Domain` as an example.
   b. Enter a short description for the category schema in the Description field.

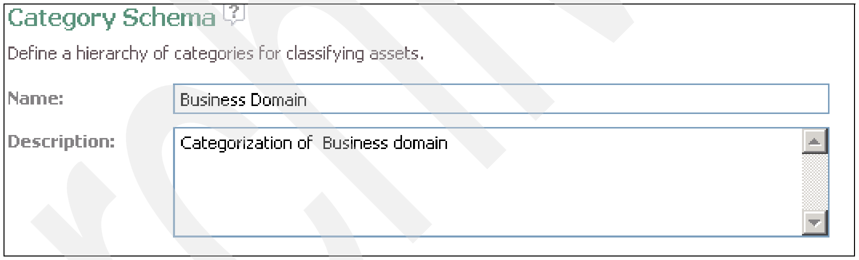


Figure 6-44 Defining category schema

5. To add a primary level of the categorization, click **Add field**.
6. In the New Category dialog box, follow these steps (see Figure 6-45 on page 159):
   a. Enter a name for the category node in the Name field.
   b. Enter a description for the category node in the Description field.
   c. Depending upon the business logic, select **Children are exclusive** to allow only one child per parent to be selected when assigning categories to a new asset.

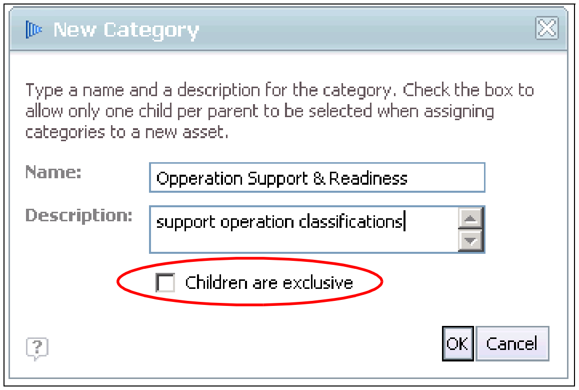


Figure 6-45 Adding primary level categorization

7. Click **OK** to add the primary category.
8. To add child categories to the category node, click **Insert Child**. Repeat this step to add as many children to a category as needed to fulfill the category schema. There is no limit to the number of categories that you can create in a schema.
9. To reorder the category nodes, use the up and down arrows next to the category nodes. Reorder the category nodes in a logical order, so that when users choose the categories, they can quickly understand and choose the correct nodes in the hierarchy (see Figure 6-46).

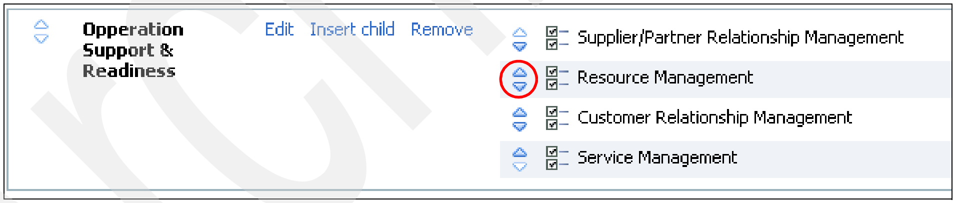


Figure 6-46 Reordering nodes in the category schema

10. Repeat steps 5 to 8 as needed to complete defining the category schema.
11. Click **OK** to save.

6.4.2 Relationship types

During the submission of an asset, users specify the relationship of the asset to other assets that are related to each other and describe the nature of that relationship. These relationships facilitate search, governance, and asset reuse.

You need to define the types of relationships in the repository so that users have the appropriate relationship types from which to choose. You are also required to define the inverse of the relationship, so the reverse relationship can be specified.

For example, *Asset_A* contains *Asset_B,* which means *Asset_B* is contained by *Asset_A.* Administration privilege is required to create an asset attribute:

1. Log on to the Rational Asset Manager Web Application as an Administrator and click the **Administration** tab.
2. In the Administration page, click **Relationship Types** in the Repository Administration sidebar.
3. Click **New Relationship** to create a new relationship.
4. In the New Relationship dialog, type the name for the new relationship type and the reverse of that relationship. For example, a relationship type named `Parent` might have a reversed relationship named `Child`.
5. Click **OK** to save.

6.4.3 Asset attributes

An *asset attribute* describes the use of an asset in relevance to a context. This metadata facilitates search, which can help other users understand how and when to use an asset. Asset attributes can be created with specific values. Administration privilege is required to create an asset attribute.

In the ITSO-Tel scenario, a service governance attribute is required in the asset repository to search for assets with defined values of service specification, service category, service version policy, and service configuration. To define this attribute and its associated values, perform the following steps:

1. Log on to the Rational Asset Manager Web Application as an Administrator and click the **Administration** tab.
2. In the Administration page, click **Asset Attributes** in the Repository Administration sidebar.
3. Click **New Attribute** to create a new asset attribute.
4. In the New Attribute dialog, define the attribute:
   a. Enter the name of the attribute, which is `Service Governance,` in the Name field.
   b. Enter a brief description in the Description field.
   c. To specify a specific value to be assigned to the attribute, select the **Use preset values** check box and type a value for each of the ITSO-Tel values

to be displayed when this attribute is present and click **Add** (see Figure 6-47).

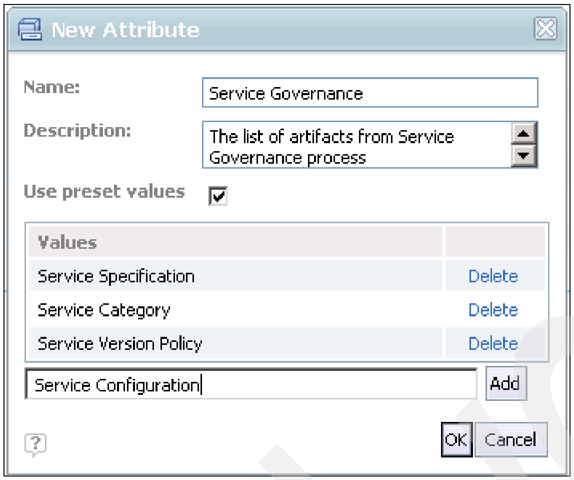


Figure 6-47 Adding a new attribute in the repository

5. Click **OK** to save.

6.5 Service discovery

Discovering services can be accomplished by searching, browsing, and evaluating services that reside in the asset repository. A *service* is an asset, but an asset is not essentially a service.

An asset consumer searches the asset repository with the purpose of discovering or finding a reusable asset and then using it in the desired context.

6.5.1 Searching assets in Rational Asset Manager

This section describes how to search the asset repository to find appropriate assets for use in ITSO-Tel projects and applications:

1. One simple approach is using the Advance search option:
   a. The asset browser or asset consumer logs in to Asset Manager and clicks the **Search for Assets** tab.
   b. Click **Advance**, which appears beside the search text box, and Rational Asset Manager will provide multiple optional parameters into which you

can feed the information about an asset to filter it out from the asset repository (see Figure 6-48).

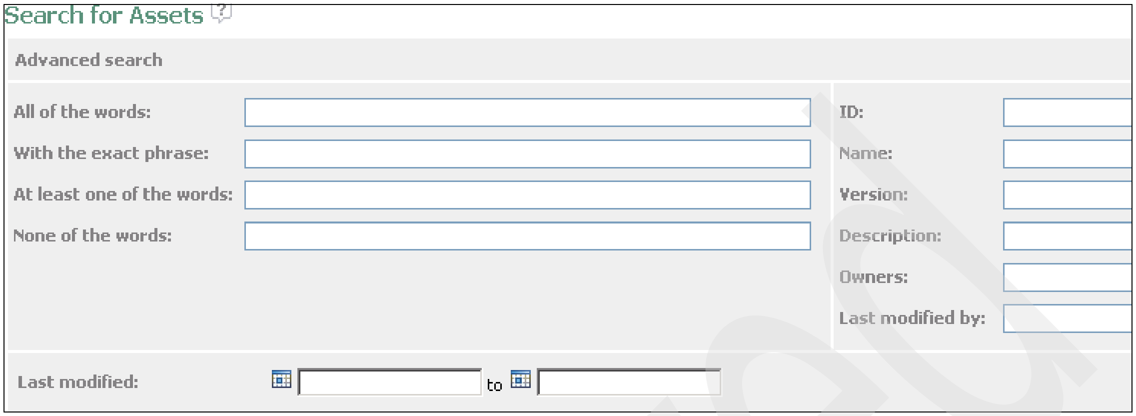


Figure 6-48 List of information parameters that can be used for advance search

2. Click **Search** to get the list of assets, depending upon how you specify the search criteria. Here, we have intentionally shown you the blank window of the options.
3. One structured approach for searching is to use **values** from category schemas. This approach implies that the assets in the repository have been classified according to the category schema. And then, these values are available as filters that can be selected and combined with other filters to create a search query. This approach is useful if the category schema is well understood by the asset consumer.

   Follow these steps to search by category schema classification:

   a. The asset browser or asset consumer logs in to Asset Manager and clicks the **Search for Assets** tab.

   b. In the Filter Your Search section, under the heading Category, select a category in which you want to search (see Figure 6-49 on page 163). Here, we have selected `SOA` category for an example.

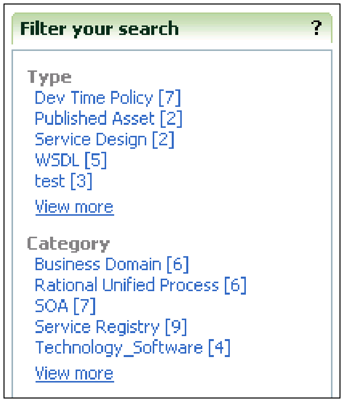


Figure 6-49 Searching by category schema classifications

c. The system will show you all of the assets that have been classified in your chosen category in the asset repository (see Figure 6-50).

Name	Version	State	Community
Account Management Business Process	1.0	Approved	Service Analysis
Account Management DB	1.0	Approved	Service Development
Account Management Service Design	1.0	Approved	Service Development
Appointment Services Process model	1.0	Approved	Workforce Management
Appointment Services Process model	1.0.1	Approved	Workforce Management
Credit Management Architecture	1.0	Approved	Service Development
Credit Management Service Design	1.0	Approved	Service Development
Credit Management Service Interface	1.0	Approved	Service Development
Credit Management Service Test	1.0	Approved	Service Testing
Credit Managment Case for Change	1.0	Approved	Service Analysis
ManageWorkforce Service model	1.0	Approved	Workforce Management

Figure 6-50 Results of search by category schema classification

4. Searching with *keywords* is a little less structured, and it does imply the keyword or something similar to it is found in the assets. When searching with keywords, the matches might be found in the asset metadata or the matches might be found in the asset artifacts. Some repositories provide fuzzy

searching capabilities, taking into consideration misspelled words or similar words.

5. A less structured style of searching is to use *tags*. With this approach, the asset consumer associates their own words with the asset. This allows the asset consumer to view one or more assets in the repository by using their own words:

   a. The asset browser or asset consumer logs in to Asset Manager and clicks the **Search for Assets** tab.

   b. A Tags section displays that reflects available tags (see Figure 6-51).

Figure 6-51 Visible tags in the Search for Assets tab

6. Click a **tag** in which you might be interested to search assets.

7. One of the challenges for asset consumers conducting searches is to browse result lists as the number of assets in the repository grows. Reducing the number of clicks to arrive at a useful, non-overwhelming result list becomes a challenge:

   a. It is possible to combine multiple search techniques into a single search query. To perform the search or to narrow down your search results, you can select search criteria available in the *Filter your Search* sidebar in the Search for Assets tab. For example, an asset consumer might use some keywords combined with several filters to create a search query, such as `Appointment` with the filters `Type: service` and `State: Approved`. This means that the repository can search for assets of type Service, in the state of Approved, containing the term Appointment somewhere in the metadata or the asset artifacts (see Figure 6-52 on page 165).

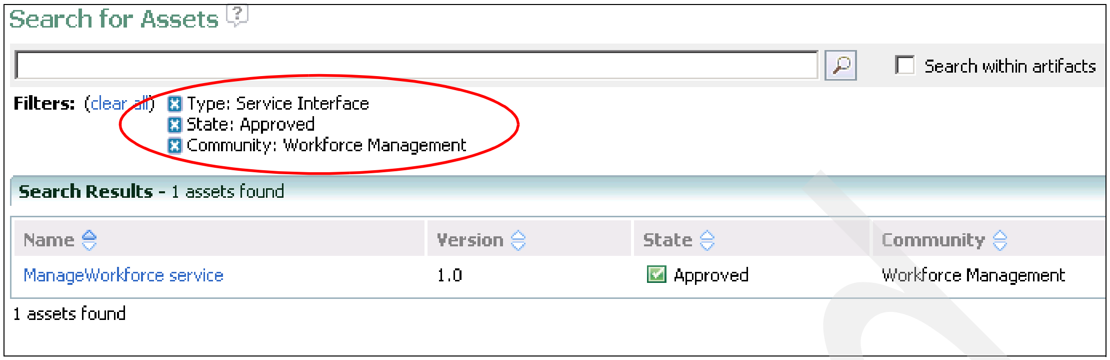


Figure 6-52 Multiple search criteria

8. From the Filters list, you can also remove any filter by clicking the X icon that appears to the left of each filter (see Figure 6-52).

6.5.2 Evaluating the Reusable Asset Specification (RAS)

How can you identify that a selected asset is a service that fits the business needs?

After finishing a search, an asset or service consumer needs to browse the asset to make sure that the identified asset matches the desired service requirement. Evaluating a service that is an asset can require technical attention and also the team's ability to use the asset successfully to meet the needs of the business.

Next, we describe the asset browsing features of Rational Asset Manager:

Select an ITSO-Tel asset from the search result list and view the asset details of by clicking the asset's name:

1. The General Details tab provides information, such as description, category, related assets, and recent activities (see Figure 6-53 on page 166).

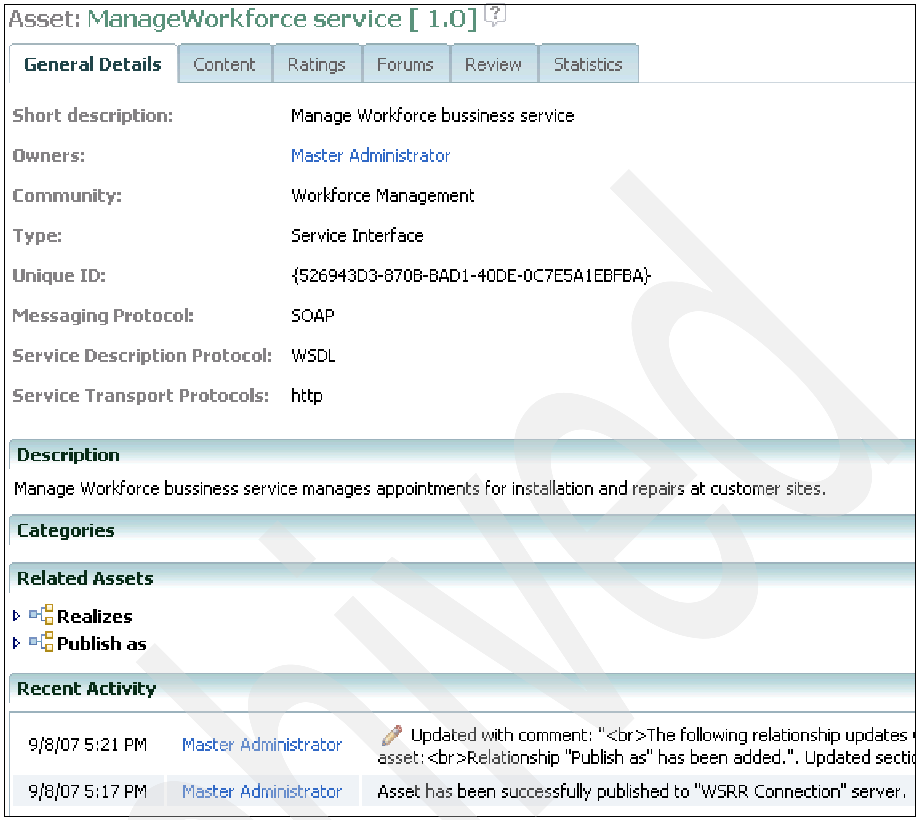


Figure 6-53 General details of a service in asset repository

2. Click the **Content** tab to browse the associated artifacts of the asset (see Figure 6-54).

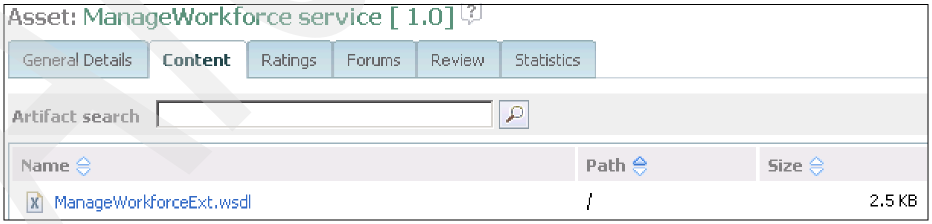


Figure 6-54 Contents view of a service in asset repository

b. Click the **Ratings** tab to browse asset feedback and discussion threads (see Figure 6-55 on page 167).

Figure 6-55 Browsing ratings of a service in the asset repository

3. Click the **Forums** tab to browse submitted defects, enhancements, or any other forum discussions associated with the asset (see Figure 6-56).

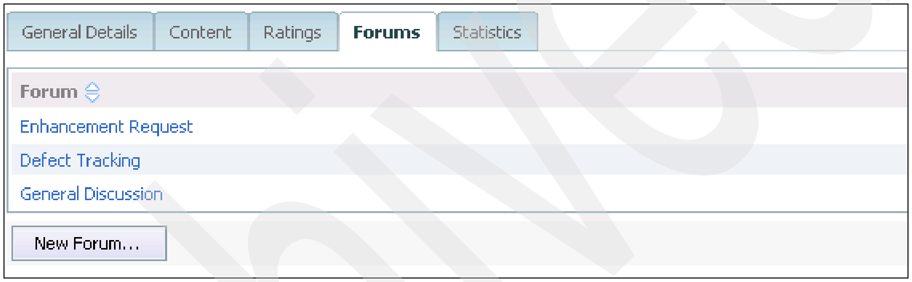


Figure 6-56 Available forums associated with a service in the asset repository

Clicking any of the forums will take the asset browser to the particular forum interface. Here, we have shown the ClearQuest Defect Tracking for an example. In this example, the asset browser can view the defects by clicking the **All Defects** query or can log a defect by clicking **New Defect** (see Figure 6-57 on page 168).

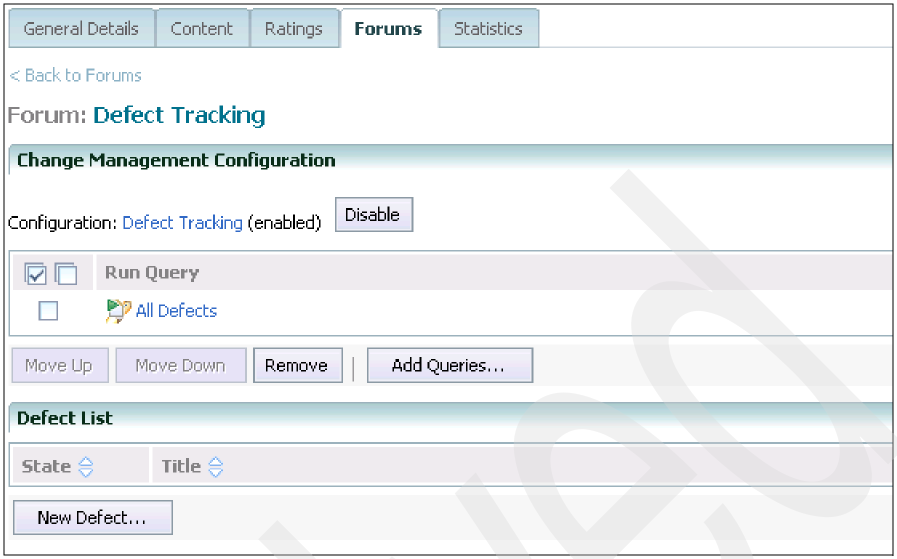


Figure 6-57 ClearQuest Defect Tracking forum associated with a service in the asset repository

4. Click the **Review** tab to conduct a review if you are authorized and the asset is under review (see Figure 6-58).

Figure 6-58 Review status of an asset

5. Click the **Statistics** tab to get a summary of the activity on the asset, such as searches and downloads:

   a. On the Statistics tab, click **Downloads** in the Reports section to see the Downloads report of an asset (see Figure 6-59 on page 169).

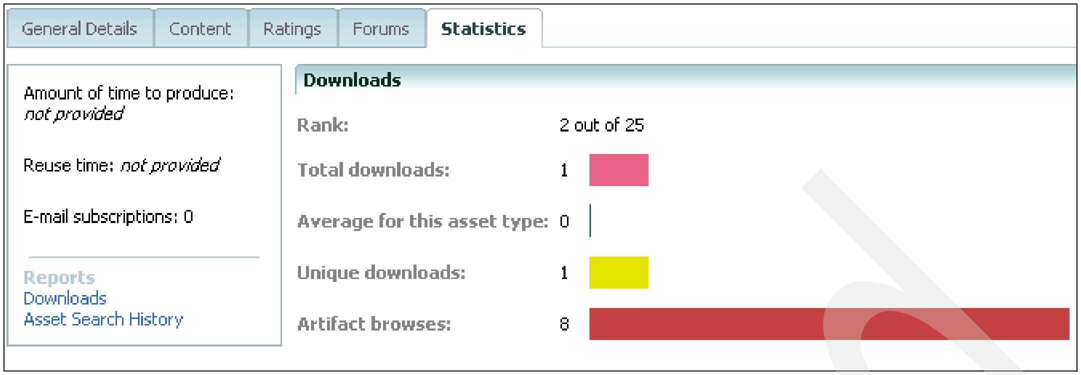


Figure 6-59 Download summary of a service in the asset repository

6. Click **Asset Search History** in the Reports section to find details about the searches made on the asset (see Figure 6-60).

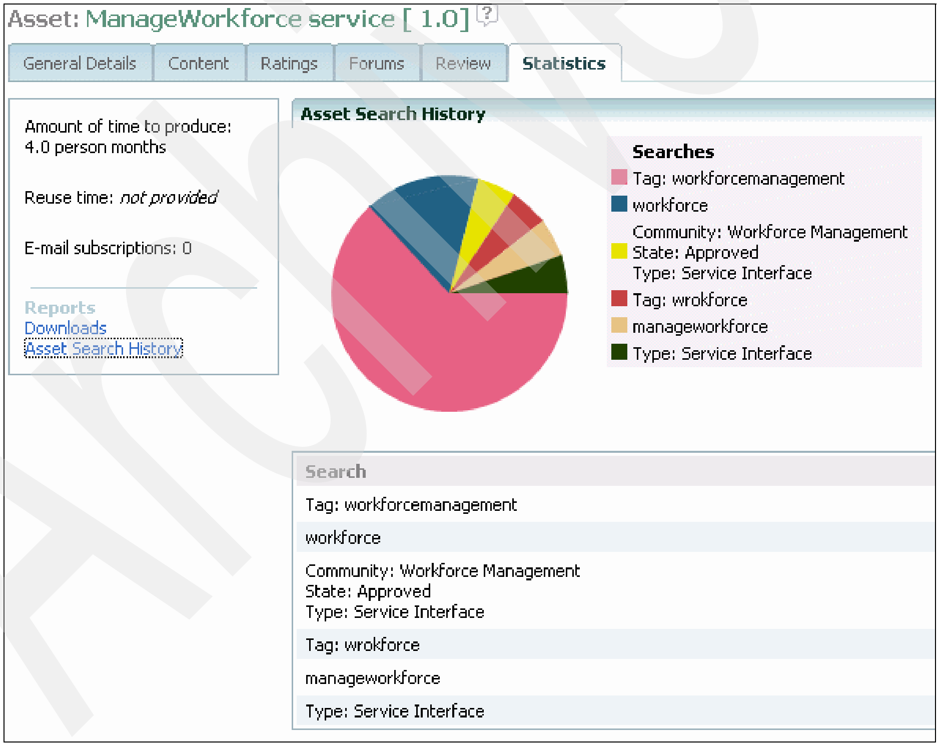


Figure 6-60 Search history of a service in asset repository

6.6 Service creation

With the inception of service development, all the artifacts should be stored in the asset repository as an asset or a group of assets. The asset specification can be defined by Rational Asset Manager metadata, which will support further reuse of the asset or defining the Reusable Asset Specification (RAS).

6.6.1 Creating an asset

An *asset* contains one or more artifacts that provide a solution to a problem for a specific context. A reusable asset is packaged explicitly to be reused in future development activities, and it meets certain criteria applicable to reusable assets.

During submission of a new asset into the ITSO-Tel repository, you must provide descriptive information about the asset (see Figure 6-61 on page 171 for example).

To begin submitting an ITSO-Tel asset:

1. Log on to Rational Asset Manager.
2. Click the **Submit an Asset** tab, which brings up the Describe page (Figure 6-61 on page 171).
3. On the Describe page of the Submit an Asset tab, enter a name for the asset. Here, we have shown creating a service named `ManageWorkforce Service` as an example.
4. Then, enter a version for the asset.
5. From the list, select a community where the asset will belong.
6. Select the asset type that best fits this asset. If there is no appropriate asset type defined, you can contact your repository administrator to create one.
7. Enter a meaningful brief description about the asset.
8. To further describe the asset, use any of the following optional tasks:
   a. Enter custom tags in the Tags field that highlight the main purpose or features of the asset. Use commas to separate individual tags and group two word tags in quotation marks, for example, “Governance Policy” or “Resource Management”.
   b. For creation time, enter a numeric value and select the unit of time that it took you to create the asset. For example, if it took you four months to create the asset, enter 4 and select person months from the list.
   c. For usage time, enter a numeric amount and select the unit of time that you estimate it will take another user to gain value from the asset. For

example, if you estimate it will take another user one year to gain value, type 1 and select person years from the list.

d. Use the rich text editor to further describe the asset. For example, you can describe the business problem that this asset solves.

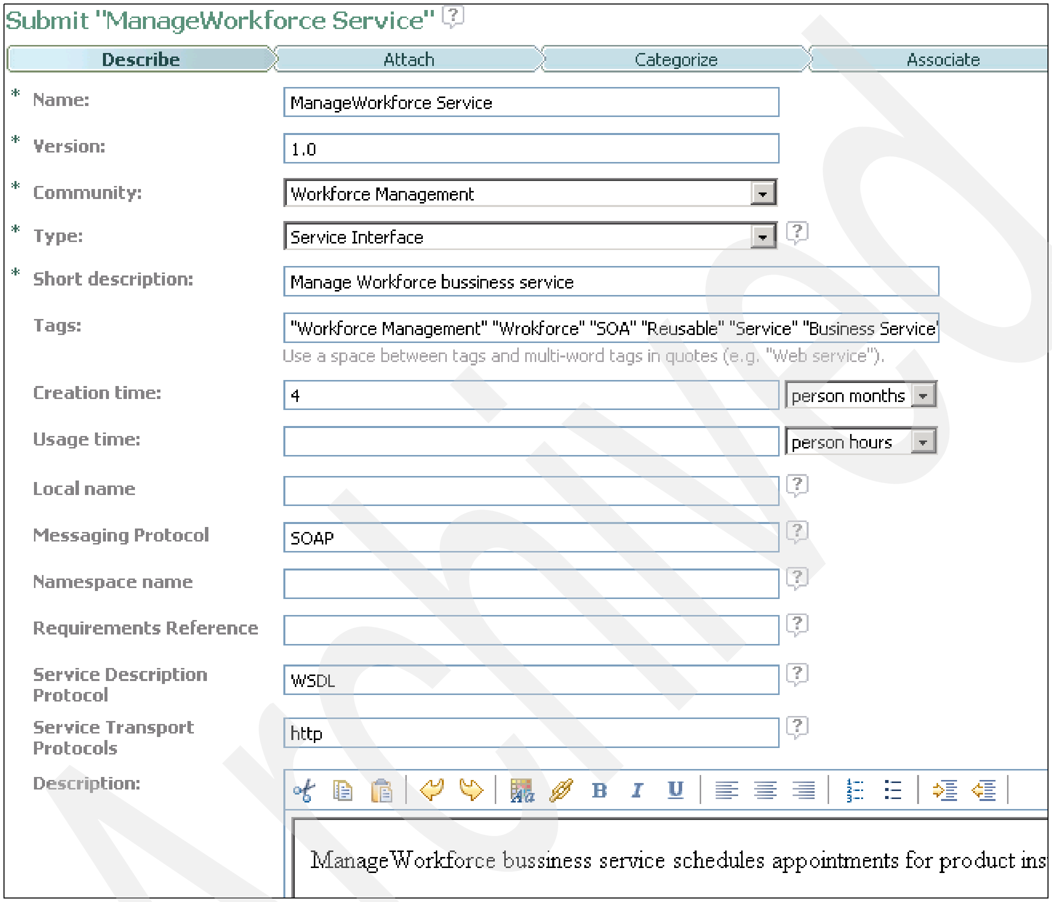


Figure 6-61 Describing an asset example

9. Click **Next** to attach artifacts to the asset.

10. in the Attach tab, click **Browse** and a Choose file dialog box will appear where you can choose the files to attach (see Figure 6-62 on page 172).

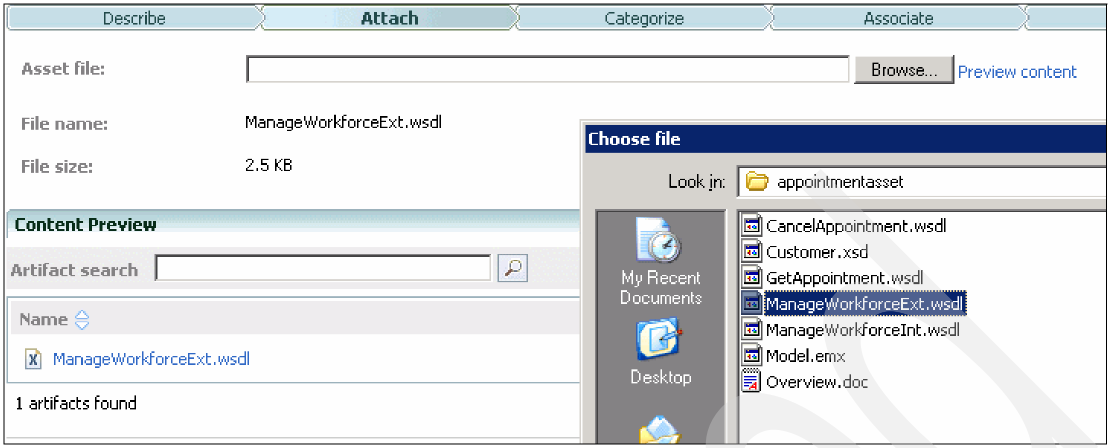


Figure 6-62 Choosing a file to attach

11. Click **Next** to categorize the asset.

6.6.2 Classifying assets through category schemas

You can classify an asset according to a reference architecture in the context of a business solution or according to the phase within a process where the asset will be used. The classification is used to capture the Asset Context in the asset metadata.

This section shows how to categorize assets using repository-level category schemas.

Successfully categorizing assets in ITSO-Tel will help users find the asset during a search. Use as many categories as possible that describe the appropriate context for the asset you are submitting. One asset can have multiple categories taken from multiple category schemas (see 6.4.1, “Category schemas” on page 156 for more information).

To categorize an asset in the ITSO-Tel repository, follow the steps:

> **Note:** We assume that you have already followed the steps mentioned in section 6.6.1, “Creating an asset” on page 170.

1. In each category, evaluate whether the asset that you are submitting fits in any of the categories.

2. Select the category and then select the subcategories from the list. These are enabled when you choose the high-level categories. Each major section represents a high-level category that your repository administrator defined.
3. Click **Add**. You can select as many categories as is appropriate for your new asset.
4. Click **Next** to specify any related assets.

Figure 6-63 Categorizing services in Rational Asset Manager

6.7 Binding related assets

Follow these steps to specify the assets in the repository that are related to the asset that you are submitting and then define the type of relationship.

ITSO-Tel assets will be listed as part of the content if they are of a relationship type that includes the new asset or contains the new asset. The reverse is also true, if the asset that you are submitting contains or includes another asset, that asset will be listed on the asset's content page.

When users search for assets, the related assets are listed in the Related Assets section. If you search and find an asset to download, the related assets are listed to make you aware of them. Downloading and evaluating all of the related assets broadens your perspective and supports successful asset reuse.

To add related assets to an ITSO-Tel asset (Figure 6-64 on page 174):

1. On the Associate page, click **Add Related Assets**.
2. In the Add Related Assets window, enter the name of the asset that you want to include as a related asset, and then click **Search**. The assets that meet your search criteria are listed in the search results view.
3. Select the assets that have a relationship with the asset you are submitting, and then click **OK**.
4. Select the relationship type between the asset you are submitting and each asset you added from the list.
5. Click **Confirm**.

6. Verify that the information has been added correctly and then click **Submit** to submit the asset.

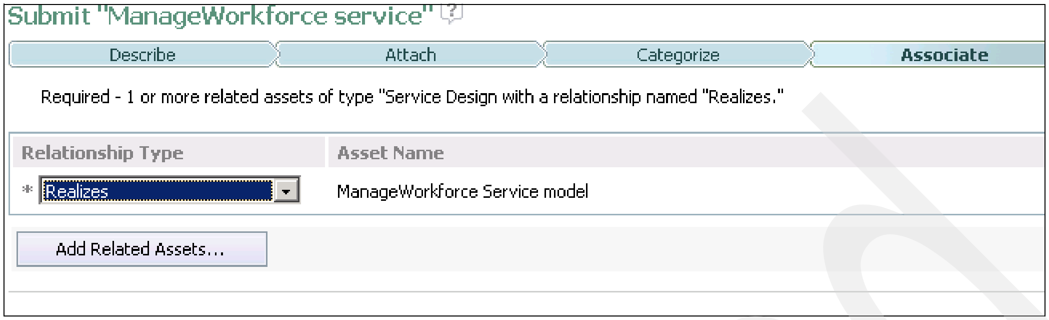


Figure 6-64   Associating related assets with relationship types

> **Note:** If there is a review process established for this asset type, click Submit As is to bypass the review process, or click Submit to review to submit the asset into review.

7. After submitting the asset, you will get options to subscribe to the asset, you can send notifications and you can also work with the associated forums (see Figure 6-65 on page 175).

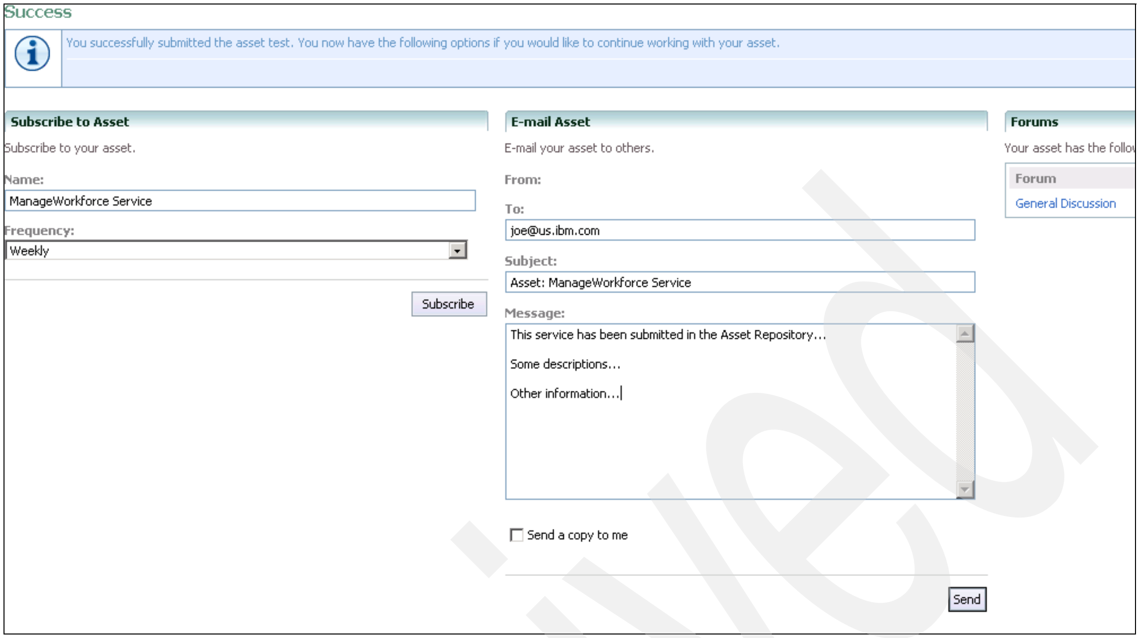


Figure 6-65 Working with the asset after submitting it

6.7.1 Asset review process

The ITSO-Tel review process applies some governance over the quality of assets submitted into the repository. The assets go through a specific review cycle when they undergo reviews as illustrated in Figure 6-66 on page 176. The gray box in Figure 6-66 on page 176 illustrates the default review cycle in the application.

There are many possible review processes given the variety of asset types, cultures, and other restrictions. The review processes are defined per community. There are three types of review processes:

- *As Is*

  This really means that there is no review process. Assets that are submitted using As Is are automatically approved. Administrators normally have to approve this case.

- *Built-in*

  In this case, the asset repository provides fundamental asset review capabilities that can be configured by the Administrator. Review Board members and reviewers can be identified to review assets of a particular type, owner, or classification. The asset repository notifies participants when an asset comes in for a review process.

- *Extensible*

  You can customize this type of review process to reflect the workflow that the enterprise needs. This review process is specified by a process language or by a change management system.

The Built-in and Extensible review processes can use a Review Board. The *Review Board* provides oversight to the review process. The Review Board members can modify the set of reviewers, initiate the review process, and cast final votes on the asset. Without a Review Board, the Reviewers are notified and the assets are approved or rejected based on the votes cast by the Reviewers.

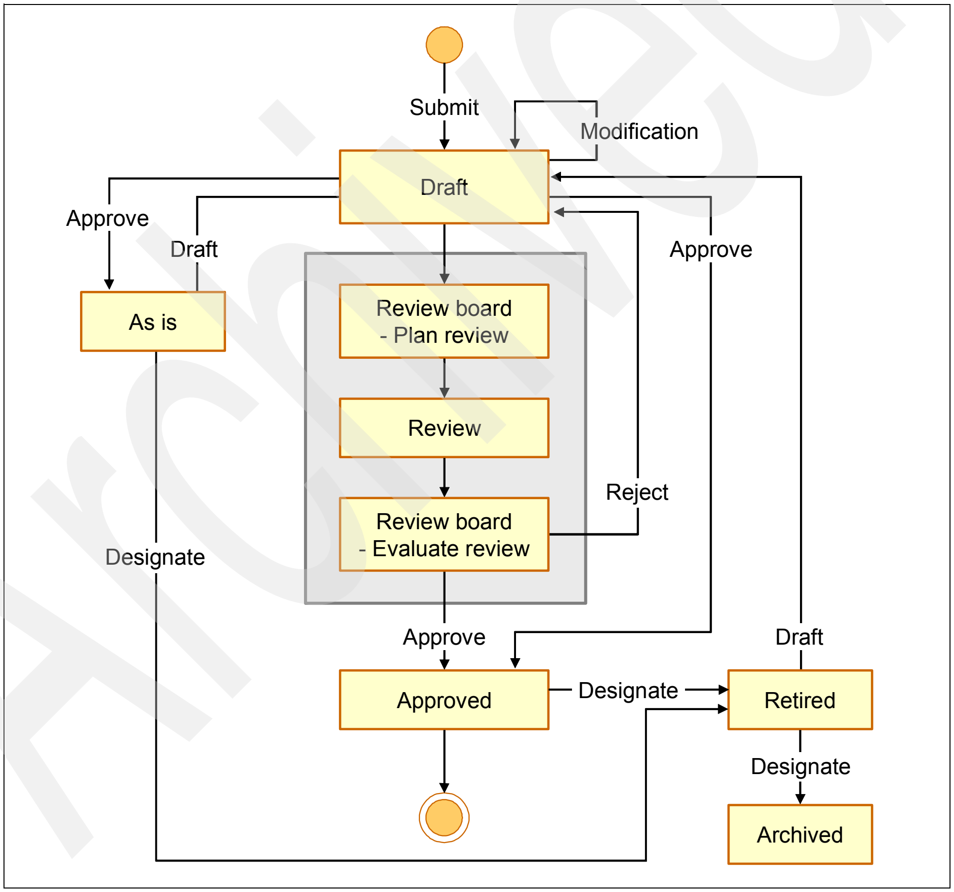


Figure 6-66 Default ITSO-Tel review process

The community administrator defines the review processes. After the asset is first submitted into a review process, the asset owner and community administrator have the ability to modify the review. This means the asset is in a *Plan review* state. Owners and administrators can add new reviewers by modifying the *threshold,* in other words, by increasing or decreasing the number of users required to review the asset. The asset owner clicks Start Review on the asset General Details page to begin the review.

After the review has been evaluated and planned, the reviewers are notified by e-mail that there is an asset awaiting their review. Users can also find a link to any asset that requires their review in the Assets to Review section of the My Asset Manager page.

Figure 6-67 shows an example of the ITSO-Tel review of the ManageWorkforce service asset.

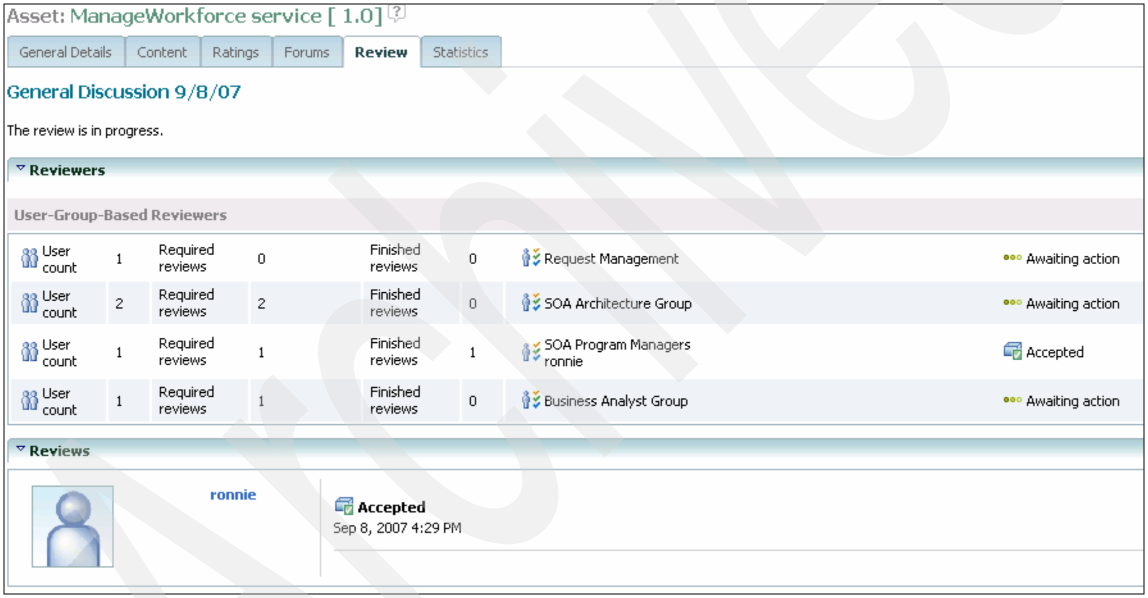


Figure 6-67 Review process history

6.7.2 Update services

We highly recommend that you familiarize yourself with the best practices for updating and creating new versions of a service in an asset repository.

You can update service elements, such as the name or descriptions, or add tags in the asset repository. However, updating service elements or adding tags can negatively affect the reusability and retrievability of the service during a search.

When you update these elements of a service in the asset repository without modifying the version number, you actually replace the existing metadata with the new variables. Because users have subscriptions to the asset's GUID, the asset subscriptions remain available; however, during a keyword search, only the updated content will be searched. For this reason, you must not change the asset's name.

You can overwrite the existing version of a service any time that you make an update to it. This is standard for smaller changes. For larger changes or updates, consider overwriting the existing version if the updates that you made fix known issues with the original service or if the updates make the service more stable in the same context in which it was created.

Next, we describe the steps to update a service or an asset in the ITSO-Tel asset repository:

1. After you have searched for the service in the asset repository, click the asset name to update it.
2. In the Asset Detail Tool sidebar, you can see all of the options as shown in Figure 6-68.

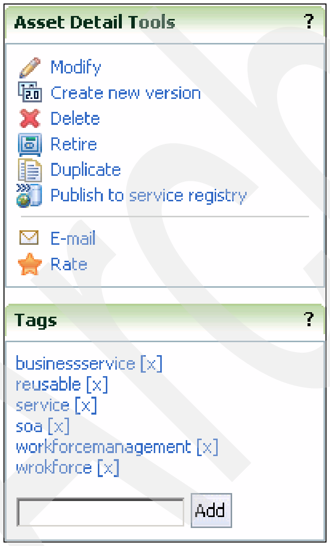


Figure 6-68 Update asset options

3. Choose between changing the service metadata (by clicking Modify) or changing the version (by clicking Create new version):
   a. Identify the ITSO-Tel location from where to check out the artifacts of the service.

b. Modify the artifacts of the service and metadata: the Owner/Producer might need to conduct merges from previous changes to the artifacts made by others that have not yet been incorporated into the published asset in the repository.

c. Submit the modified asset. Make changes to metadata and supporting documents and identify the label or tag that needs to be applied to this version.

d. The service is now be ready for the review and test process.

4. The Delete option is performed by the owner or the administrator in the asset repository. The Delete option removes the asset from the ITSO-Tel Repository. Certainly, this needs to be handled with care.

   Key considerations:

   – In most cases, the enterprise must avoid deleting the services from the repository. The exception is deleting services, which are new assets that have just been submitted to the repository and are still in the initial state or draft state (in other words, the service has *not* gone through the review and approval processes and is not available to consumers). You can delete these services from the repository.

   – Asset owners can generally delete non-approved services, which are less important assets. Services, which have been approved and are being reused, are more difficult to delete. One of the more challenging aspects to removing services is ensuring that you have communicated this to the entire set of consumers. Having support from the repository to automate much of this communication is a critical feature.

5. To Retire a service, an administrator sets the state of the service to *retired* in the asset repository. The effect is that the service is no longer visible to the consumers, but the service is still visible to the asset owner and the administrator.

6. Create a Duplicate of an service. Duplicating the service in the asset repository creates a new copy of the service, including the categorization, related assets, and relationships. The name of the service/asset will be listed as “Copy of *AssetName”* and the following note is automatically generated in the description of the duplicated asset with the asset name as a link to the original asset.

   Note: This asset was initially created with the same information as *AssetName.*

7. You can modify any of the details, content, relationships, or categorization at any time. To submit the service after duplicating it, make any necessary changes using the Submit Wizard and then submit.

8. *Tags* are specific keywords attached to a service as metadata to facilitate the search. To add a Tag, enter a word or a phrase (multiple words must be within quotation marks) in the text box in the Tags section of the Asset Details Tools sidebar. Then, click Add. You can add as many tags as you want.
9. Subscription to a service. A consumer or interested users can subscribe to a service to get notified about review comments. They can also subscribe to a forum to get e-mails about discussion topics (see Figure 6-69).

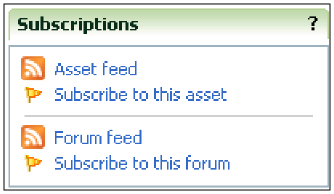


Figure 6-69 Subscribing to a service or forum

6.8 Service publication

This section discusses publishing a service to the ITSO-Tel WebSphere Service Registry and Repository.

6.8.1 Overview of service publication

In this task, a service is made visible to a target audience that goes beyond Rational Asset Manager. As part of the normal review activities, the service can transition to the state of *Approved*. It is then visible to the consumers with the appropriate access control.

Here, we are interested in making the service visible to other target audiences. However, this task is applicable to services in the asset repository, such as Service Interfaces, which are published into service registries.

In this case, the Service Interface likely has several key artifacts, such as the WSDL and XSDs. At this point, the approved Service Interface asset is selected and the ITSO-Tel production target service registry is identified. The contents of the Service Interface (or whatever asset type it is) asset are published to this service registry.

This can affect the state of the service in the asset repository by transitioning to another state, such as *published*. Publishing the asset introduces issues around classification and security. The classification of the asset in the asset repository might not map directly to the classification that is available in the target repository or registry.

You must also consider security and access control. In certain cases, the access control to the ITSO-Tel WebSphere Service Registry and Repository can be dissimilar. It is critical that one repository does not betray the access control specified in another repository.

ITSO-Tel's development-time repository is focused on the governance of assets to aid the development organization. The runtime service registry is focused on the governance of services into and out of production environments (Figure 6-70).

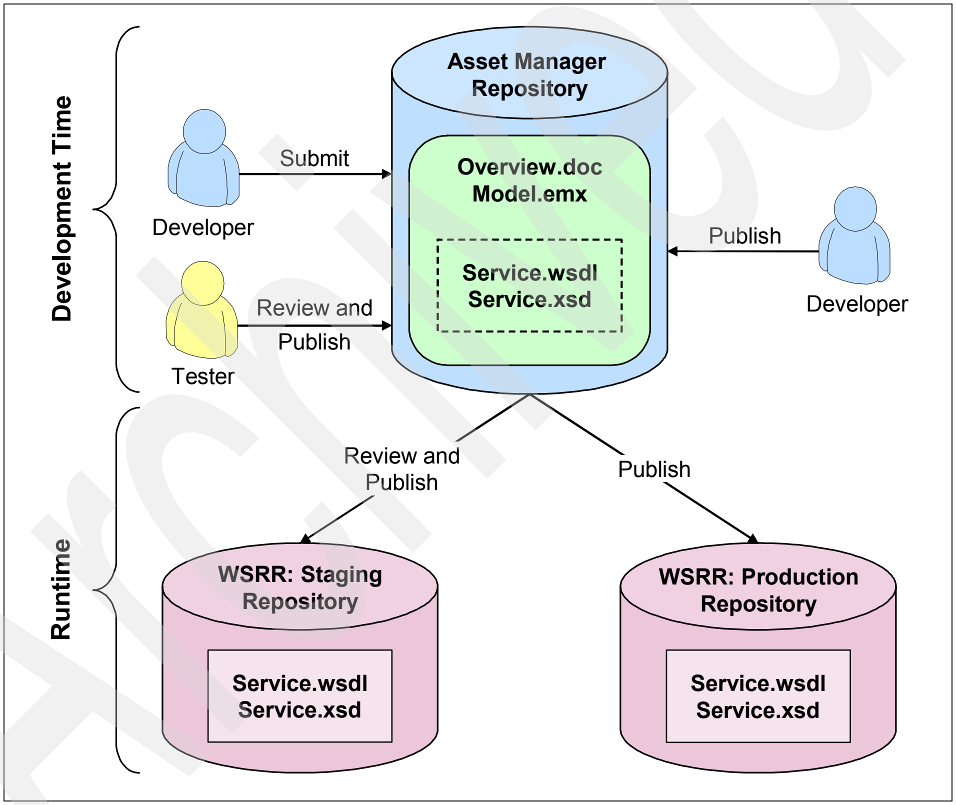


Figure 6-70 ITSO-Tel's development-time and runtime repositories

6.8.2 Publishing services to WebSphere Service Registry and Repository

Assets that contain Web services files can be published to ITSO-Tel's WebSphere Service Registry and Repository directly from the asset repository.

Before services can be published, administrators must configure the URL for the ITSO-Tel production WebSphere Service Registry and Repository server and create a connection for the community in which the service resides. Services for publication must be in the *Approved* state before they can be published. Only one service can be published at a time.

To publish, the owner or an administrator clicks **Publish to service registry** in the Asset Detail Tools sidebar of the asset's information page (see Figure 6-71).

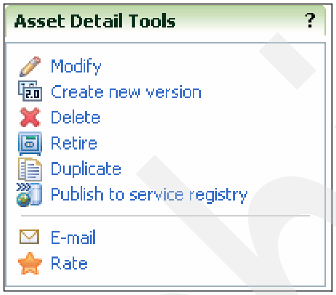


Figure 6-71 Publish to service registry link

Selecting services to publish

You can select the artifacts from an asset to publish to ITSO-Tel's production WebSphere Service Registry and Repository (WSRR):

1. On the Select page of the Publish wizard, the artifacts in the asset that can be published to WebSphere Service Registry and Repository are listed according to the folder structure within the asset. The file types that can be published include XSD, WSDL, module, policy, and XML.
2. Select the artifacts to publish. To select all of the artifacts, select the check box next to the name of the asset (see Figure 6-72 on page 183).

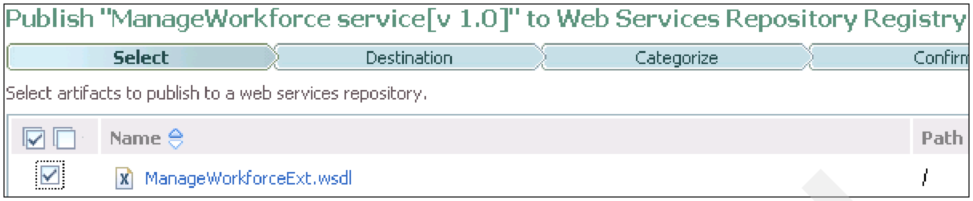


Figure 6-72 Selecting artifacts to Publish to WSRR

3. If you select any artifacts that have a dependency on other artifact types, you must also select the other artifact types on which the artifacts depend.
4. Click **Next**.

Determining the WebSphere Service Registry and Repository server

You can select from established connections when publishing an asset. If no connection is available or if you do not see the connection that you want to use in the list, request that your community administrator create a connection.

On the Destination page of the Publish wizard, specify which connection to use when publishing the asset. The connection determines to which ITSO-Tel WebSphere Service Registry and Repository server the asset will be published:

1. Select an ITSO-Tel WebSphere Service Registry and Repository connection from the list.
2. If the WebSphere Service Registry and Repository server has security enabled, type a user name and password to use when publishing the asset. The publish record will be filed according to the user name provided here.
3. To retain the user name and password with this asset, select **Save credentials for future use** (see Figure 6-73).

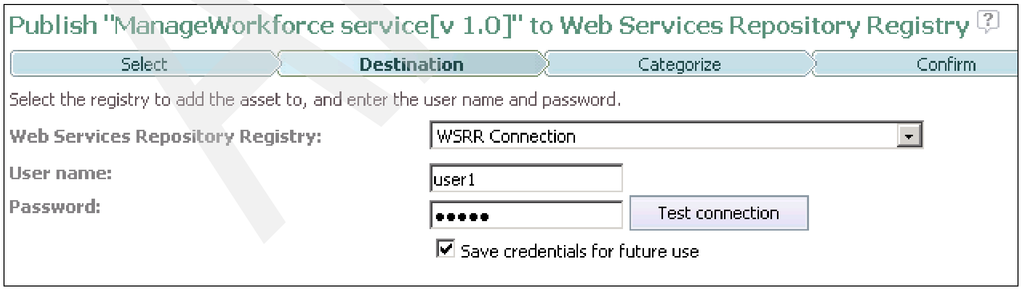


Figure 6-73 Selecting a WebSphere Service Registry and Repository connection

4. Click **Next**.

Categorizing assets based on WebSphere Service Registry and Repository classifications

You can specify how assets will be categorized according to the category schema on the ITSO-Tel production WebSphere Services Registry and Repository server:

1. The categories shown are those defined on the ITSO-Tel production WebSphere Service Registry and Repository server. You can choose from those categories listed to categorize the asset to be published. These categories will apply to the asset in the Web services registry.
2. To categorize the service, select a category from a list and click **Add**. Repeat to select as many categories from the available lists as you want (see Figure 6-74).

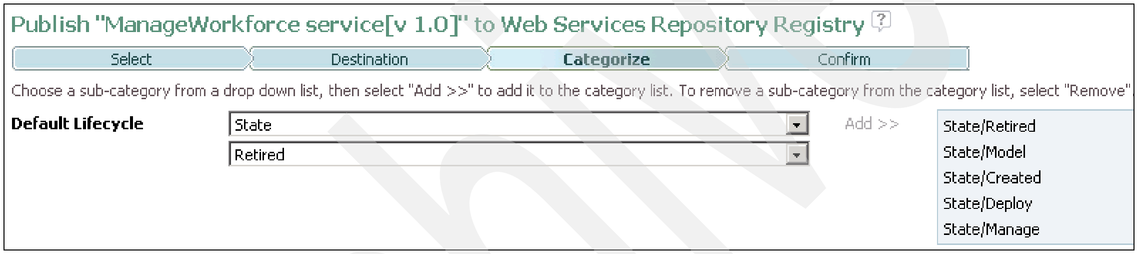


Figure 6-74 Categorizing the service for publication to WebSphere Service Registry and Repository

3. Click **Next**.
4. The Confirm page will appear. Click **Publish** after you verify the details. The Service will be published to WebSphere Service Registry and Repository.

Verifying published services in WebSphere Service Registry and Repository

After you have published the service from Rational Asset Manager to the ITSO-Tel WebSphere Service Registry and Repository, navigate to the Related Assets section in the General Details tab. You will be able to see the service under the *Publish by* subhead. Click the Service name:

1. The General Details page will display and you will be able to see a Link to the ITSO-Tel production WebSphere Service Registry and Repository. Click that link to go to the WebSphere Service Registry and Repository (see Figure 6-75 on page 185).

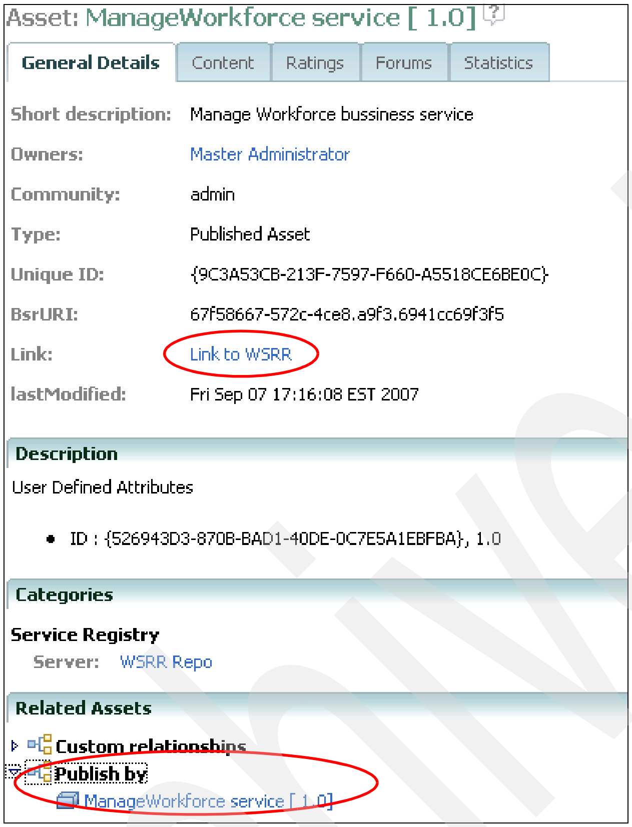


Figure 6-75 Published service link to WSRR

2. In WebSphere Service Registry and Repository, you will be able to see the service, description, and relevant information about the asset (see Figure 6-76 on page 186).

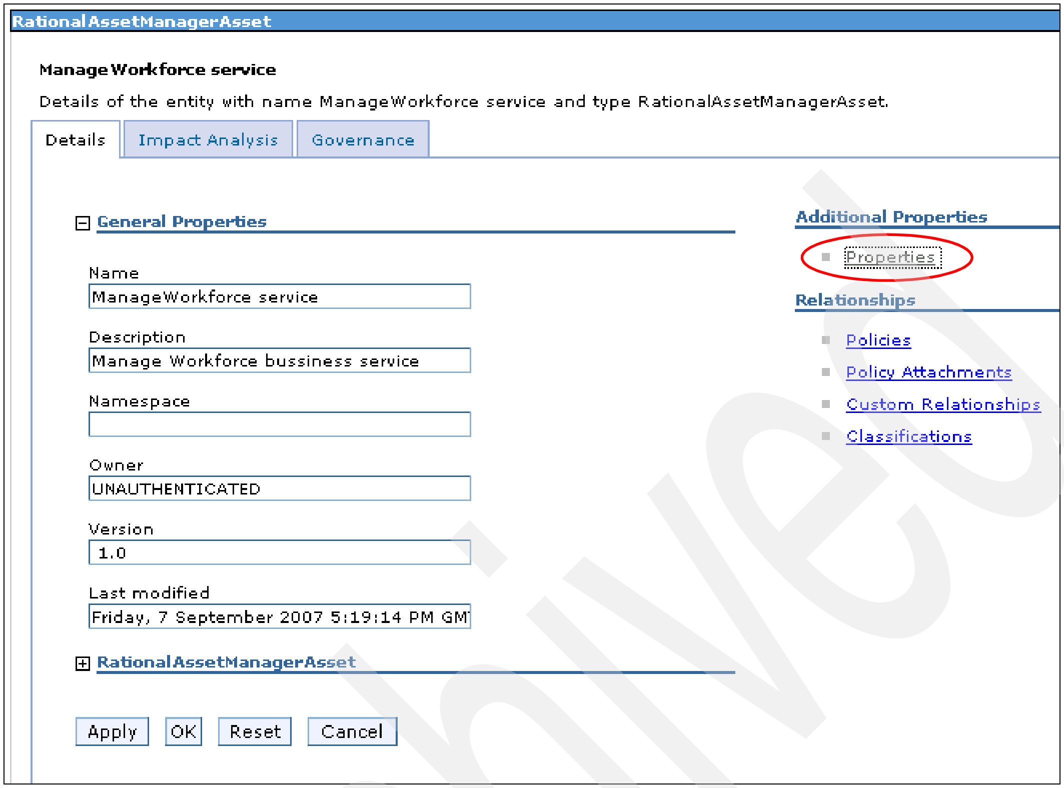


Figure 6-76 Published service details in WebSphere Service Registry and Repository

3. Click **Properties**, which is circled in Figure 6-76, to see additional properties. There, you will be able to see the WebSphere Service Registry and Repository metadata as well as the AssetManagerLink, which points to the asset in the Rational Asset Manager repository (see Figure 6-77 on page 187).

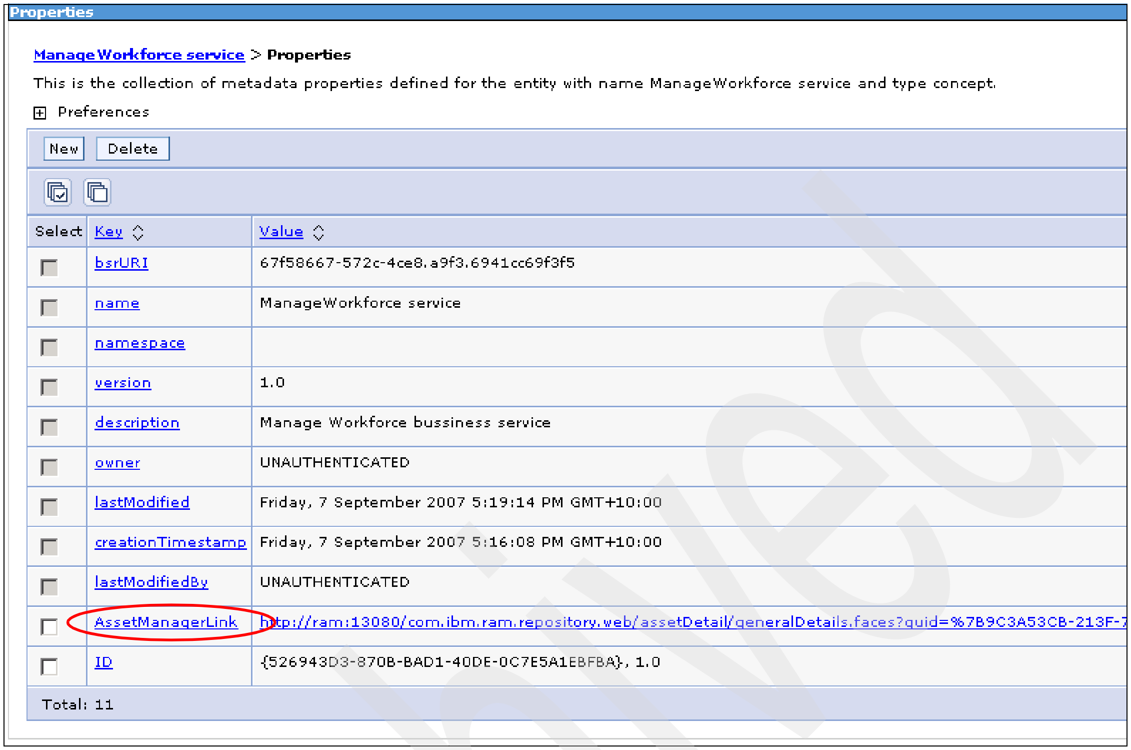


Figure 6-77   Asset Manager asset link in WSRR

6.9 Evaluating asset repository usage

The ITSO-Tel repository administrators can access reports to evaluate repository usage in Rational Asset Manager.

The repository contents, asset or service activity, and user activity reports use the statistics index. The schedule for the index builder can be configured on the Configuration page. Each report can be downloaded as a comma-separated values (CSV) document.

Some of the reports are tracked by the repository, and other reports are tracked by other tools or roles. These metrics come from various sources, such as consumers, asset owners, producers, or reviewers, the asset repository, or from other users.

Asset or service consumer metrics generally include qualitative metrics, such as reusability ratings (such as 4 out of 5 stars) or text feedback in the form of comments and discussions. Typically, these metrics are stored in the asset repository.

Service Owners, Producers, and Reviewers tend to create asset-level metrics. These metrics include the level of effort for a service consumer to use the asset and the level of effort to create a specific version of a service in the asset repository. The metrics also include size, such as the number of bytes, lines of code, or number of components. Typically, these metrics are stored with the asset in the repository.

Repository metrics are largely quantitative. Repository metrics include metrics, such as the number of downloads for a specific service version, the number of searches and types of searches, the number of browses from searches, as well as information about which service consumers conducted certain downloads. Other metrics can be repository activity over a specific period of time, such as the number of repository accesses, submissions, and so forth over a certain period of time. The repository can also track the number of problems or defects associated with a service.

In general, repositories do not track the cost side of service development. This is typically tracked in project management and financial reporting tools.

The following information will help you use and evaluate the repository-level reports for ITSO-Tel.

In the Administration page in the Repository Statistics sidebar, you can find the following reports mentioned and described along with an aggregation report that is controlled by the "Show statistics from" list (see Figure 6-78):

Figure 6-78 Repository Statistics sidebar

- **Repository Contents**: This report shows the contents of the ITSO-Tel repository, such as assets by community, assets by type, and assets by state. You can expand or collapse each section as needed (see Figure 6-79).

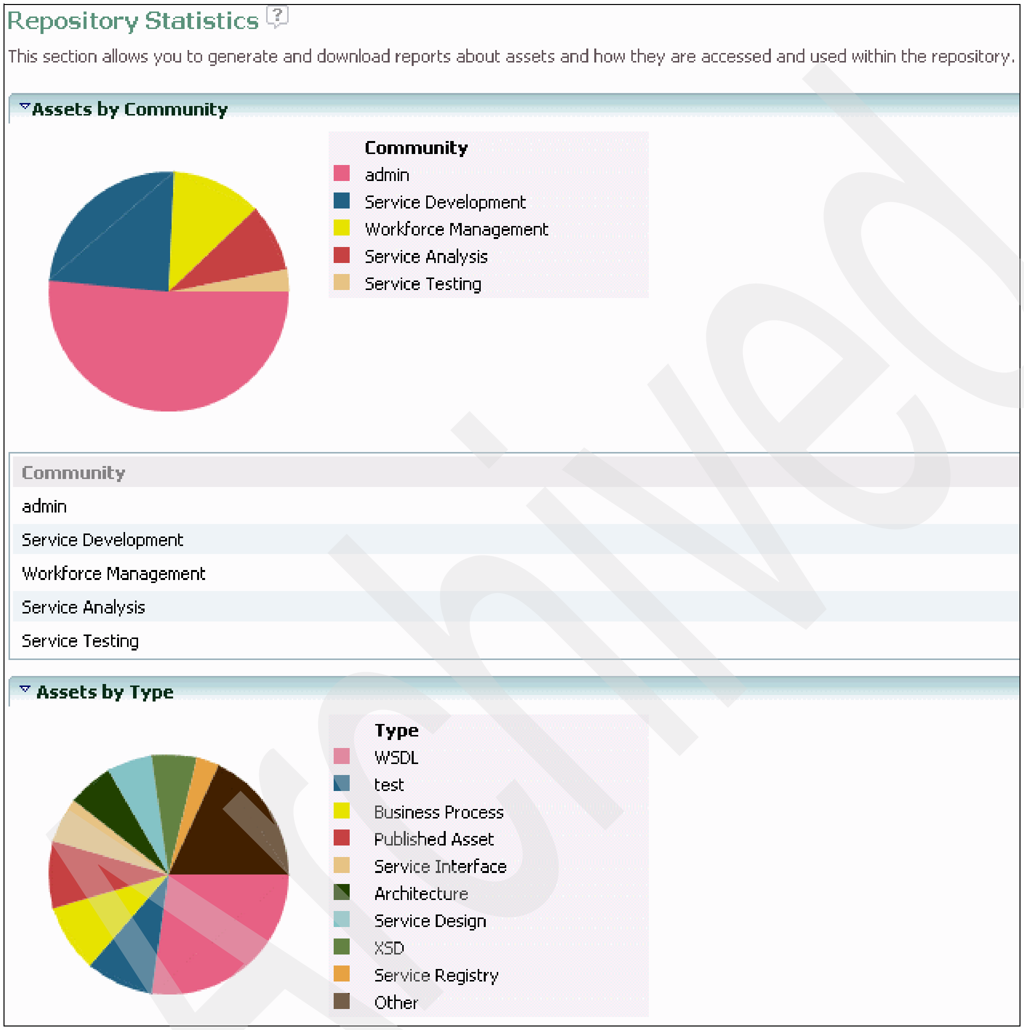


Figure 6-79 Part of Repository Contents report

- **Asset Activity**: This report mainly shows how many times a particular version of an asset has been downloaded, subscribed to, or, in other words, the number of consumers or interested parties.
- **User Activity**: This report reflects the number of users logging into the system, depending upon the time frame chosen in the “Show selected from” list on the Repository Statistics sidebar.

- **Search History**: This report shows the search activity of users over a selected period of time. The Search Count column of the report is the number of times that the search was performed in the selected time frame. The Search Views column is the number of times assets were viewed from the search. This report is useful, because it shows what users are actively looking for in the repository. If the search count is substantially smaller than the search views, it might mean that users are not finding what they are looking for when performing searches, and they click several of the results per search. If the search count is substantially larger than the search views, it means that users are performing searches and not viewing the results, possibly because there were too many results and none of the top results looked valuable, or because there were too few results.
- **Activity Audit**: This report provides a list of activities where single or multiple activities can be selected for a certain time period to generate a report. Simply select the list activity check boxes, specify the date range in the From and To boxes, and click the Generate Audit button (see Figure 6-80 on page 191).


Repository Statistics

This section allows you to generate and download reports about assets and how they are accessed and used within the repository.

Activity Audit

	Name	Description
☐	License acquired	A user received a license.
☑	Asset state change	An asset changed state, either as a result of a submis
☐	New review instance	An asset entered into review.
☐	Review	A reviewer reviewed an asset.
☐	Reviewer rating	A reviewer rated an asset.
☑	User rating	A user rated an asset.
☐	Anonymous rating	A user rated an asset and chose not to display his or
☐	New forum	An asset owner or administrator created a new forum
☐	New forum topic	A user created a new topic.
☐	New forum post	A user responded to a topic.
☐	Search view	A user viewed an asset as a result of a search.
☐	Asset download	A user downloaded an asset.
☐	Asset subscription created	A user subscribed to an asset.
☐	Asset subscription deleted	A user deleted a subscription to an asset.
☑	Asset submit	A user submitted an asset.
☑	Asset update	A user modified an asset.
☐	Asset delete	A user deleted an asset.
☐	Task submit	The system or a user created a task.
☐	Task complete	A user marked a task as completed.
☑	Asset type created	A repository administrator created an asset type.
☐	Asset type modified	A repository administrator modified an asset type.
☐	Asset type deleted	A repository administrator deleted an asset type.

From: Aug 1, 2007 To: Aug 31, 2007 Generate Audit

Figure 6-80 Activity Audit generator page

7

Dynamic assembly and run time

This chapter describes how to build a runtime repository for service discovery using WebSphere Service Registry and Repository and a process runtime environment using WebSphere Business Services Fabric.

This chapter uses the fictional organization ITSO-Tel (as described in Chapter 4, “Our business scenario” on page 87) to illustrate the use of WebSphere Service Registry and Repository and WebSphere Business Services Fabric. We use the Manage Workforce service and its two interfaces, GetAppointment and CancelAppointment, as examples in this chapter.

This chapter contains the following sections:

7.1 Introduction

In this chapter, we use WebSphere Business Services Fabric with WebSphere Process Server as the process execution platform. We also use WebSphere Service Registry and Repository as the repository for all corporate services that will be replicated into the WebSphere Business Services Fabric Business Service repository.

We use WebSphere Integration Developer with the Business Composition Studio to submit changes to WebSphere Business Services Fabric that will impact runtime execution.

Figure 7-1 shows the products used in this chapter and the relationships among them.

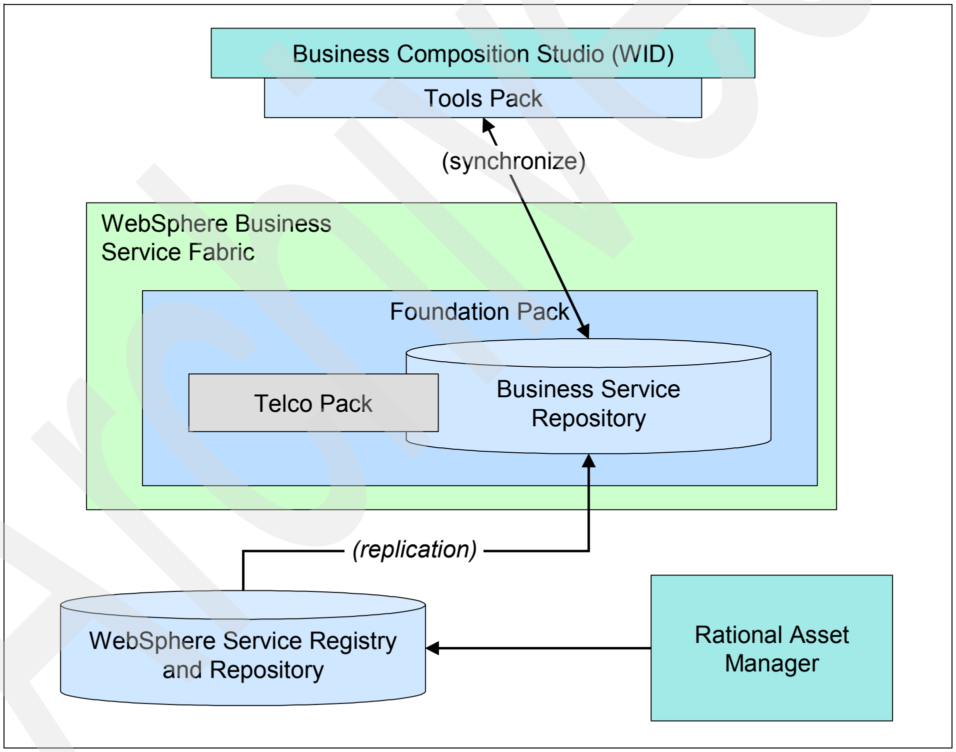


Figure 7-1 Relationships among products

Business Composition Studio submits and receives all changes made from and into WebSphere Business Services Fabric. WebSphere Business Services Fabric will receive information from WebSphere Service Registry and Repository every time that a new service (such as a Web service) is loaded into WebSphere Service Registry and Repository. The service information is copied into WebSphere Business Services Fabric when replication occurs.

In our scenario, we follow these steps:

1. Rational Asset Manager publishes several files for the Workforce Management service (WSDL and XSD) into WebSphere Service Registry and Repository.
2. We create a project in WebSphere Business Services Fabric that connects to WebSphere Service Registry and Repository in order to retrieve all its services (including the ITSO-Tel ones).
3. From WebSphere Integration Developer, synchronization occurs with WebSphere Business Services Fabric in order to retrieve the up-to-date WebSphere Business Services Fabric project image.
4. We add composite services, applications, and application suites to the project.
5. Finally, we submit these changes into WebSphere Business Services Fabric and deploy the processes to run.

7.2 Publishing services

During the life cycle of any service-oriented architecture (SOA) solution, new services, common applications, and processes are developed. Sometimes, services are built and at other times, they are reused. In all of these situations, there is a step in which each service is published into a corporate repository. In this section, we discuss the publication of services into a corporate repository running in WebSphere Service Registry and Repository.

In this section, we address two mechanisms for publishing services into WebSphere Service Registry and Repository:

- Through Rational Asset Manager

  Rational Asset Manager is used as an asset repository, which stores many types of assets, including services. Service assets can be published to WebSphere Service Registry and Repository.

- Manually

  If services are not registered as assets in Rational Asset Manager, we can publish them directly to WebSphere Service Registry and Repository.

WebSphere Business Services Fabric also contains a repository for services. The only mechanism to publish services to the WebSphere Business Services

Fabric Business Service Repository is through WebSphere Integration Developer.

7.2.1 Publishing from Rational Asset Manager to WebSphere Service Registry and Repository

As seen in Chapter 6, “Service life cycle management using Rational Asset Manager” on page 117, Rational Asset Manager is the central asset manager, which manages all services as assets.

Details about publishing services from Rational Asset Manager to WebSphere Service Registry and Repository are provided in 6.8.2, “Publishing services to WebSphere Service Registry and Repository” on page 182.

WebSphere Service Registry and Repository classification systems can be synchronized with Rational Asset Manager category schemas to classify or control life cycle states. Governance life cycle states match the states specified in WebSphere Service Registry and Repository. The WebSphere Service Registry and Repository default life cycle contains the following states:

- Initial state

  When services are not governable, there is a default transition and state for all of them.

- Created state

  To make a service governable, move it into a Created state.

- Model state

  When you are ready to plan and model a service, move it to the Model state by using the *Plan* transition.

- Assemble state

  A model for a service needs to be approved by a role, such as a Business Analyst, before work can begin constructing the service. Use the *Authorize development* transition to move to the Assemble state.

- Deploy state

  After a service has been assembled, it must be reviewed before it can be deployed. Use a role, such as an Infrastructure Architect, to complete the *Certify* transition to move to the Deploy state.

  When in the Deploy state, services are ready to be deployed, you typically deploy them first to a test environment and then to a production environment.

Whatever environment is used, there are two possible transitions for this state:

- The service must be repaired, because errors are found. In this case, the *Repair* transition is used to return the service to the Assemble state.
- The service is approved; therefore, the service is running and the *Approve* transition is used to move to the Manage state.

► Manage state

In the Manage state, services are controlled by management systems. The following situations can occur:

- The service is acceptable, so nothing happens and it remains operative.
- New business requirements have led to the development of a new version of the service. In this instance, use the *Deprecate* transition. This moves the service to the Retired state.
- The service is not meeting some of its Service Level Agreements (SLAs). In this instance, the *Revoke* transition is used to return the service to the Deploy state.

7.2.2 Publishing directly into WebSphere Service Registry and Repository

Because WebSphere Service Registry and Repository is intended to be a corporate service registry (in our scenario, for the entire ITSO-Tel company), its console and plug-ins will be used to publish services. The concept of services in WebSphere Service Registry and Repository is not strictly associated with WSDL files, but it specializes in them.

In our scenario, we have defined that every asset must be managed through Rational Asset Manager, so it is important to realize that this mechanism does not reflect this new service in a Rational Asset Manager repository.

Every service in WebSphere Service Registry and Repository that is being governed must have WSDL associated with it. Therefore, WebSphere Service Registry and Repository is intended to store all corporative WSDL files and the services that they represent.

These are the steps involved in publishing services manually into WebSphere Service Registry and Repository:

1. First, access the WebSphere Service Registry and Repository console through `http://<hostname>:<port>/ServiceRegistry` (default values for WebSphere Service Registry and Repository are `localhost` and `9080`,

assuming that you are on the same machine on which WebSphere Service Registry and Repository runs, but in this example, we are using port `9083`).

2. Expand **Business Metadata** → **Business Services view,** click **Business Services**, and then click **New** as in Figure 7-2.

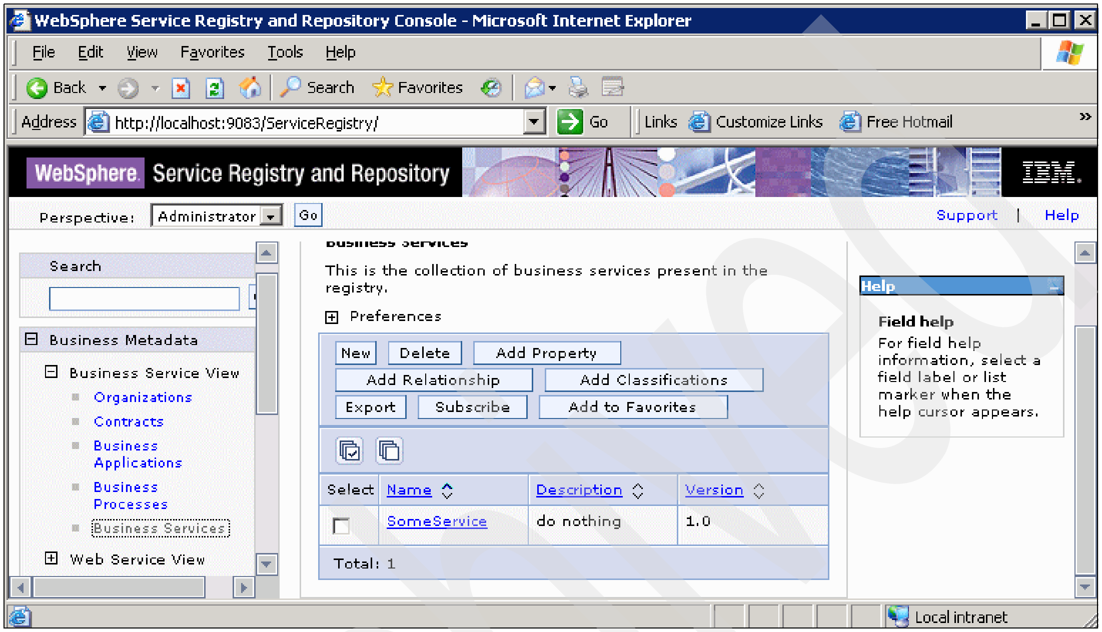


Figure 7-2 Creating a new service in WebSphere Service Registry and Repository

3. Create a simple service and enter all of the basic information about the new service. In this example, we create a new service named `CustomerAccount`, so click **CustomerAccount** in the Business Service view.

4. In this section, you can define more detailed information about the service. Also, three tabs are shown: Details, Impact Analysis, and Governance. Click **Governance** as shown in Figure 7-3 on page 199.

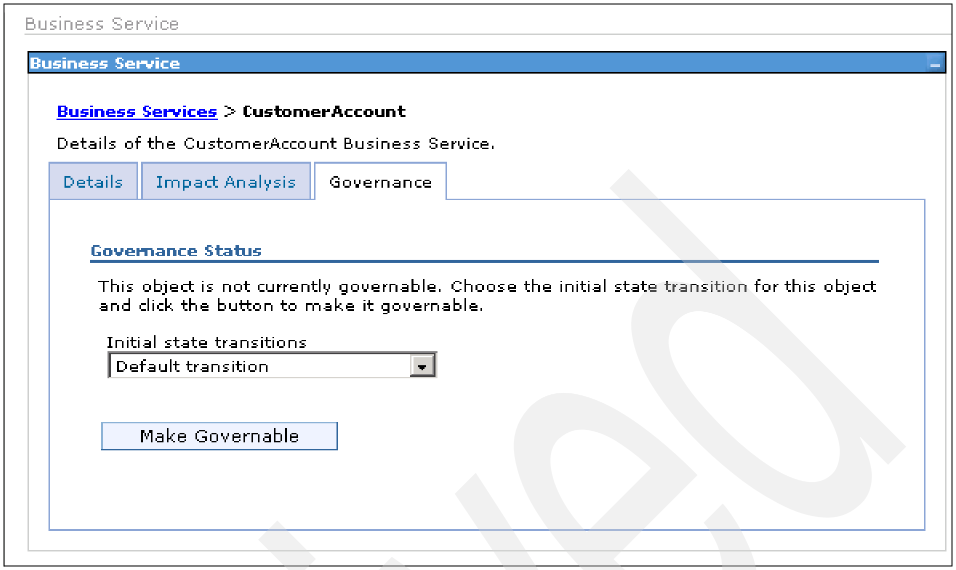


Figure 7-3 Making a service governable

5. When we decide to govern this service (by clicking **Make Governable**), many types of validation occur from WebSphere Service Registry and Repository in order to control the whole service life cycle. Then, many types of concepts can be related to this service:
   - Business organizations, applications, process, contracts, and so forth
   - Web services, such as Ports, Interfaces, and Policies
   - All kinds of documents, such as XML and XSD
6. Click the **Details** tab. From here, we can specify (Figure 7-4 on page 200):
   - The interface of the service (the WSDL port type) by clicking **Provided interface**
   - The available endpoints for the service (the WSDL ports) by clicking **Available endpoints**

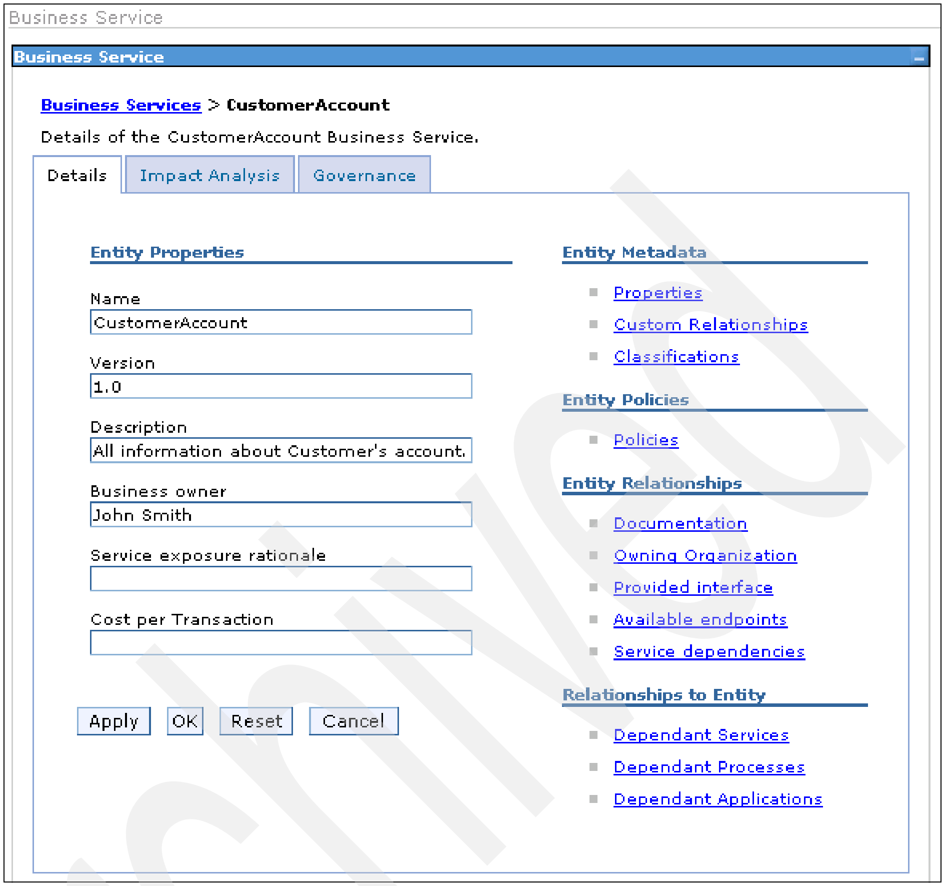


Figure 7-4 Assign interface and endpoints to services

7.2.3 Synchronizing with WebSphere Integration Developer

WebSphere Business Services Fabric provides plug-ins for WebSphere Integration Developer to manage integration projects.

Therefore, we must create a workspace in WebSphere Integration Developer in order to store all WebSphere Business Services Fabric synchronized projects. After the workspace is created and WebSphere Business Services Fabric is installed and configured, we can create a new project.

> **Note:** For information about how to install WebSphere Business Services Fabric, refer to Appendix B, “WebSphere Business Services Fabric installation” on page 301.

We will create a project for our fictional company, ITSO-Tel:

1. Open the WebSphere Business Services Fabric console. We used the following URL, logging in with the default user ID `admin` and the password `webify`:

   `http//localhost:9080/fabric/app`

2. Create a new project:

   a. From the console, go to **My Services** → **Governance Manager** → **Configure Projects**, and click **Create a project**.

   b. Enter the following information (Figure 7-5) and then click **Create project**:

      i. Project Name: `ITSO-Tel`

      ii. Description: `ITSO-Tel Company`

      iii. Project Type: **Business Services**

      iv. Team Organization: **System**

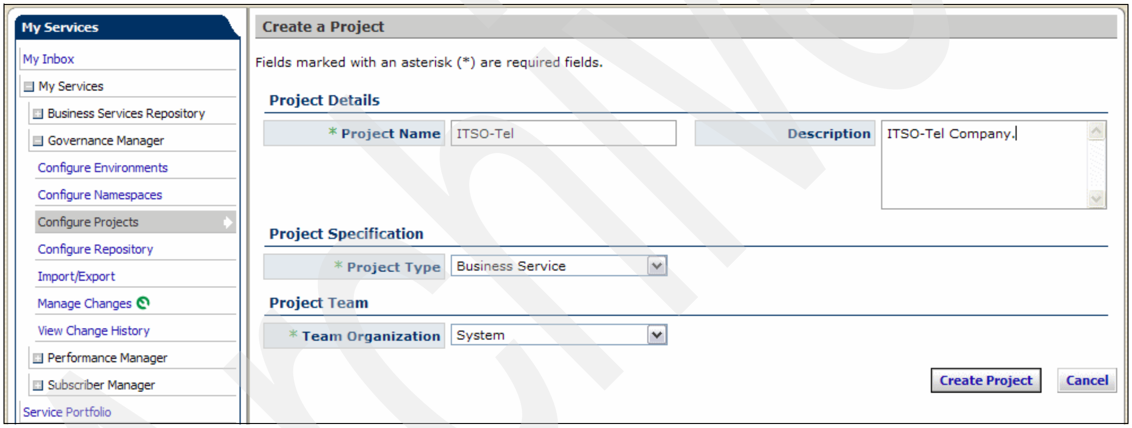


Figure 7-5 Creating projects in WebSphere Business Services Fabric

3. Create new namespaces for this project:

   a. From the console, go to **My Services** → **Governance Manager** → **Configure Namespaces** and click **Create Namespace**.

   b. Enter the following information (Figure 7-6 on page 202) and then click **Create Namespace**, which will create a new namespace for your project:

      i. Display Name: `ITSO-Tel Enrollment NS`

      ii. Namespace Type: **Enrollment**

      iii. Namespace URL: `http://itso-tel.ibm.com/enroll#`

      iv. Owner Project: **ITSO-Tel**

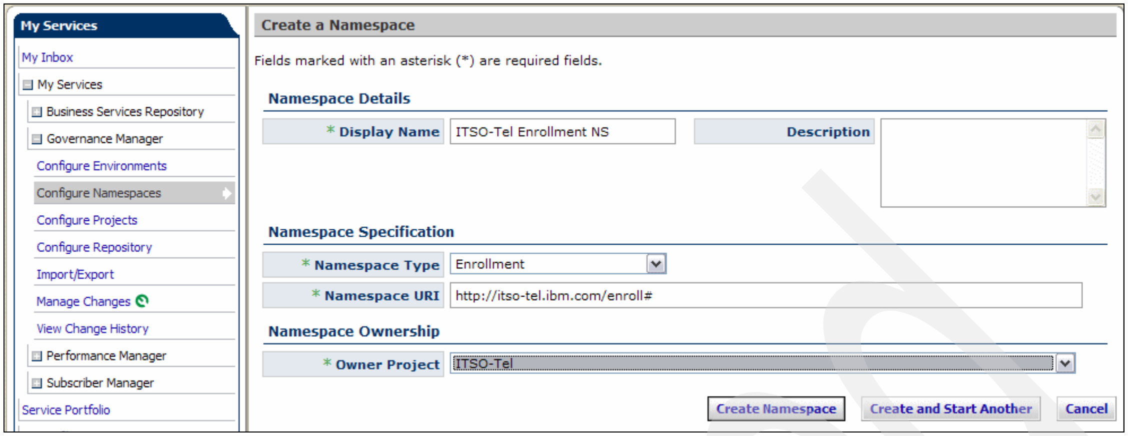


Figure 7-6 Creating namespaces in WebSphere Business Services Fabric

4. Using the same procedure, create two additional namespaces:
   a. Create one namespace with the name of `ITSO-Tel Instance NS` and a Namespace Type of **Instance**.
   b. Create a second namespace with a name of `ITSO-Tel Schema NS` and a Namespace Type of **Schema**.
5. Select the project **ITSO-Tel** and click the **Namespaces** tab to see the namespaces that you have defined (Figure 7-7).

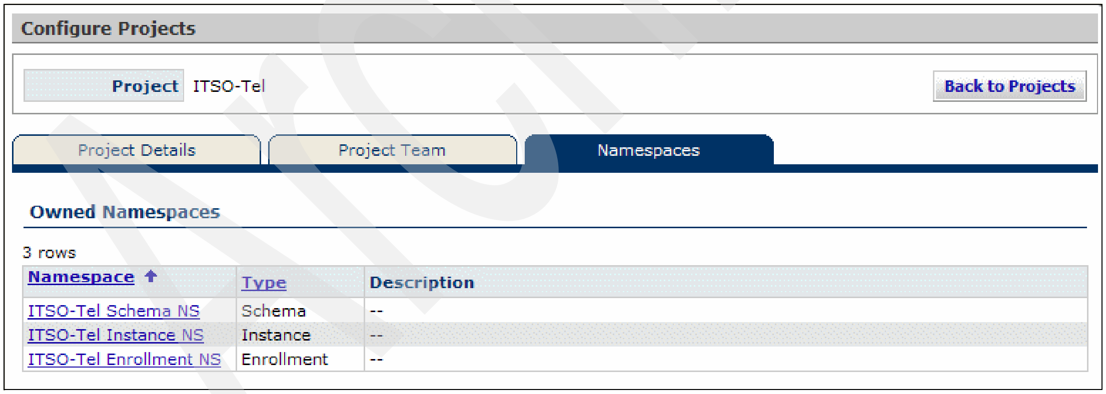


Figure 7-7 List of namespaces for the ITSO-Tel project

6. We want to reuse as much as possible from the Telecom Operation Package. Therefore, we will import a namespace from that project into our project:
   a. Click **Import Namespaces**.

b. Select all namespaces from the **Telecom Operations Core Model** project (part of the Telecom Operation Package). As described in 4.3, “Business model” on page 90, ITSO-Tel is based on the SID model (as well as eTOM), so we are reusing these namespaces based on SID.

c. Click **Import Namespaces** to import these namespaces.

7. We will now configure the replication process with WebSphere Integration Developer. When we create a project in WebSphere Business Services Fabric, we can replicate it in a WebSphere Integration Developer workspace. After it is replicated, a local copy of this repository exists in the workspace, and a user can work with this local copy. After the user has created their business services, the user submits that information to the Governance Manager. The governance administrator approves and publishes the changes in the Business Service Repository. After the changes are published, the user updates their project to retrieve the latest committed changes from the Business Service Repository into their workspace.

   Create a new workspace in WebSphere Integration Developer. We created a workspace with the directory `E:\workspaces\itso-tel`.

   > **Note:** WebSphere Integration Developer does not have to be installed on the same machine as WebSphere Business Services Fabric.

8. In WebSphere Integration Developer, create a new Fabric Project:

   a. Click **File** → **New** → **Project**.

   b. Expand **Business Services Fabric** and select **Fabric Project** (Figure 7-8). Click **Next**.

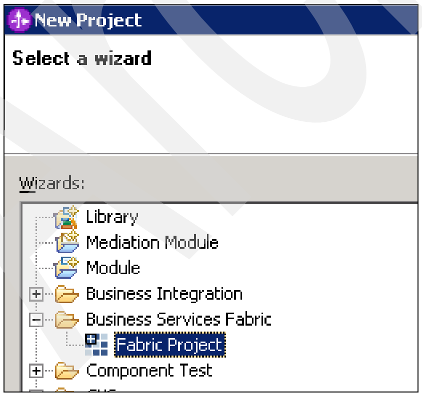


Figure 7-8 New Project in WebSphere Integration Developer

   c. Enter a Project name of `ITSO-Tel Fabric` and click **Next**.

d. Click **Configure** and then enter information to connect to the WebSphere Business Services Fabric Business Services Repository location. Enter the information appropriate to your system, and then click **OK**. A window similar to Figure 7-9 will be shown that will replicate with the repository.

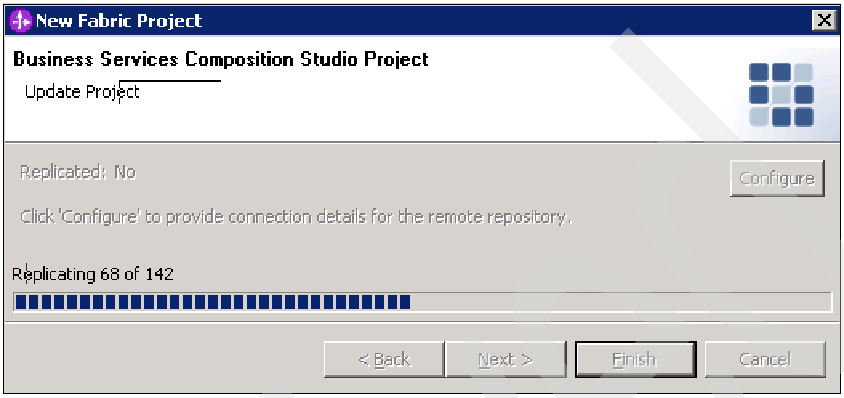


Figure 7-9 Synchronizing with WebSphere Business Services Fabric

9. When replication has finished, click **Next**. Select the Fabric project that you have just created, **ITSO-Tel**, as shown in Figure 7-10 and click **Finish**. Both projects are now synchronized.

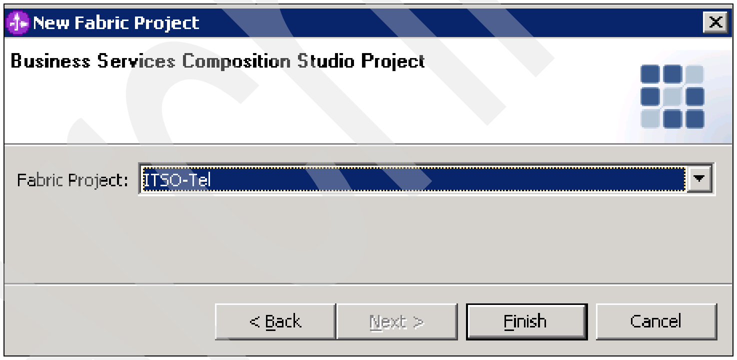


Figure 7-10 Selecting ITSO-Tel project to synchronize

10. Now we can use Composite Studio to source and assemble business services. We can work with entities, such as:

   – Application Suites
   – Applications
   – Business Services:
      • Process Service
      • Visibility Service

- Optimization Service
– Composite Services
– Interfaces
– Endpoints

11. In our scenario, we have decided to implement a business service based on the `eTOM` activity `Manage Workforce`, as described in 4.3.2, “Enterprise information model” on page 92 and 4.3.3, “Process model” on page 94. The entities that we have created are shown in Figure 7-11.

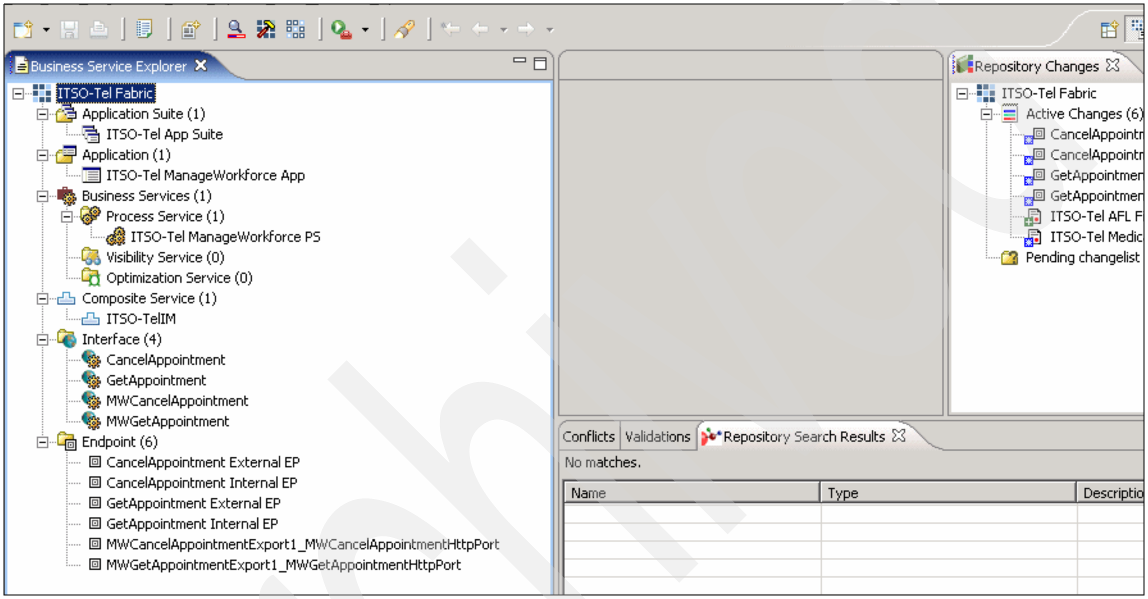


Figure 7-11 Fabric Project view in WebSphere Integration Developer

7.3 Consuming services

In an SOA solution, services are often consumed by business processes that implement composite services. However, it is very important to differentiate one particular kind of service: those that do not have a defined endpoint.

These services are called *business services* and they are important at the highest level of SOA maturity as defined in the Open Group Services Integration Maturity Model (OSIMM). At this level of maturity, services are dynamically reconfigurable and do not have a single endpoint. The endpoint definition has to be resolved at run time, depending on context, content, and policies defined for the service.

For more details about OSIMM, see:

http://www.opengroup.org/projects/osimm/

7.3.1 Services in WebSphere Service Registry and Repository

WebSphere Business Services Fabric has the capability of accessing WebSphere Service Registry and Repository as an external service repository. Creating an *External WSRR project* allows WebSphere Business Services Fabric to consume services definitions from WebSphere Service Registry and Repository.

To set up WebSphere Service Registry and Repository integration in WebSphere Business Services Fabric:

1. First, open the Fabric console in your Web browser. In our environment, we used:

   `http://localhost:9080/fabric/app`

2. Create a new External WSRR project:

   a. From the console, go to **My Services** → **Governance Manager** → **Configure Projects** and click **Create a Project**.

   b. Enter information about the project, such as (Figure 7-12 on page 207):

      i. Project Name: `ITSO-Tel Enterprise Services`

      ii. Description: `All ITSO-Tel Enterprise Services stored in WSRR`

      iii. Project Type: **External WSRR**

      iv. Source Name: `ITSO-Tel-WSRR`

      v. Sponsored NS URI: `http://itso-tel.ibm.com/spn#`

      vi. Team: **System**

   c. Click **Create Project**. This will create a new project in WebSphere Business Services Fabric.

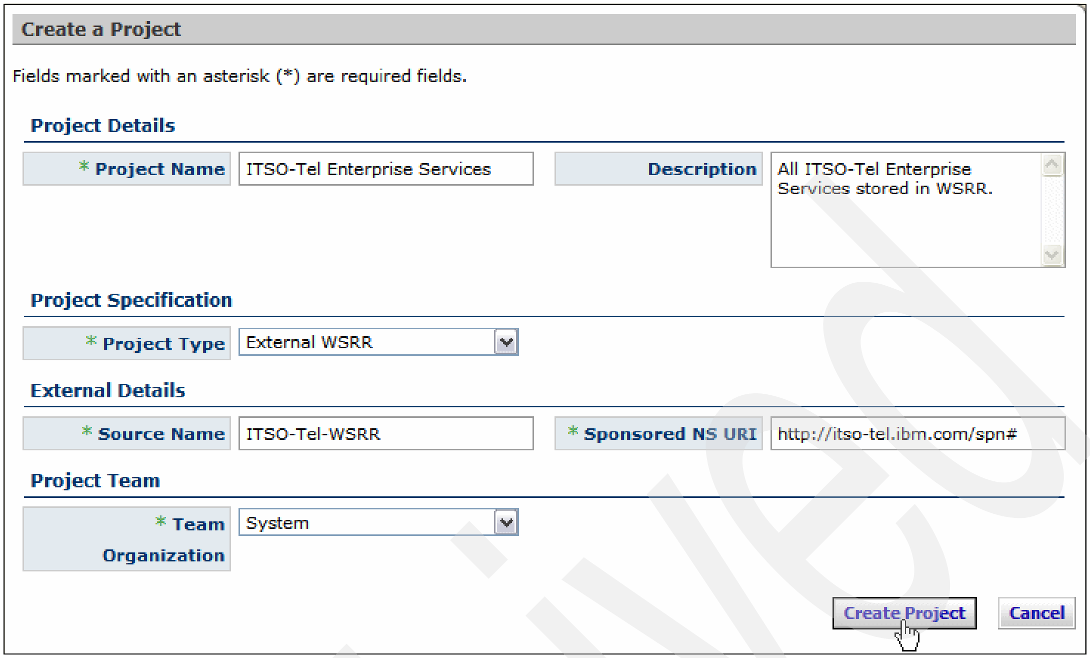


Figure 7-12 External repositories in WebSphere Business Services Fabric

3. Set the connection details and expiration settings for the External WSRR project (Figure 7-13 on page 208). When complete, WebSphere Business Services Fabric Business Service Repository should now be able to connect with WebSphere Service Registry and Repository and replicate:

   a. Click the **Federation Settings** tab.

   b. Provide the Connection Details of where WebSphere Service Registry and Repository is listening.

   c. Determine whether the connection needs to be secure by setting the Requires Security field to true or false.

   d. Click **Save**.

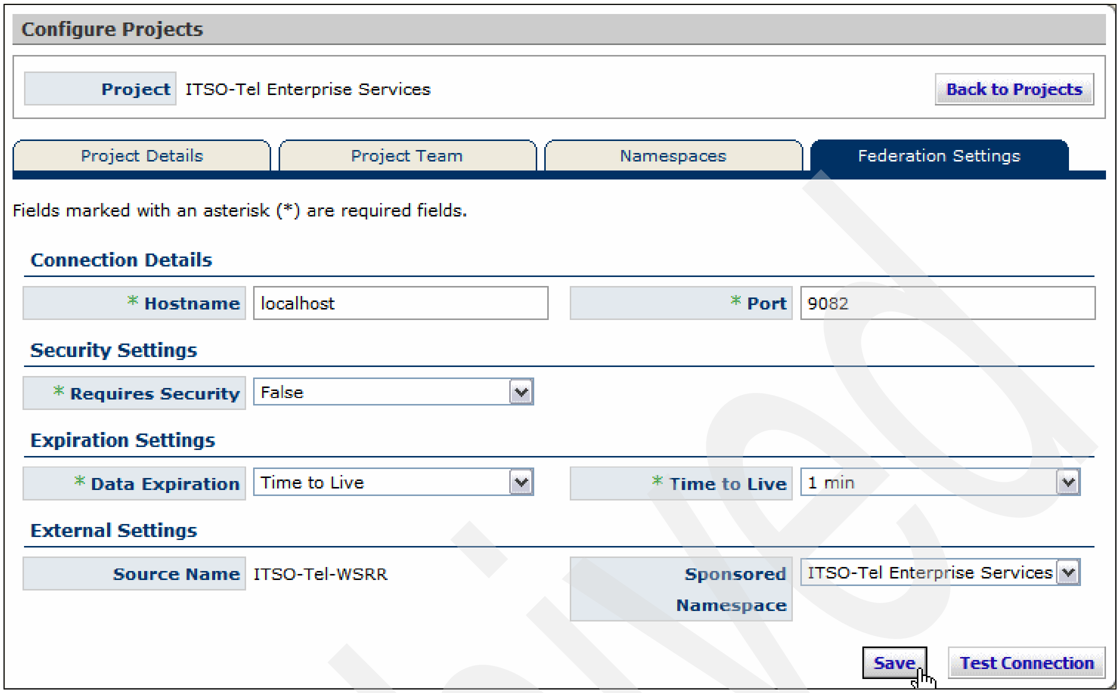


Figure 7-13 Federation Settings in WebSphere Business Services Fabric

4. You can now search for services stored in WebSphere Service Registry and Repository from **My Service → Business Services Repository**.

   Figure 7-14 on page 209 shows a search for interfaces in WebSphere Service Registry and Repository that contain `appoint` in the Name.

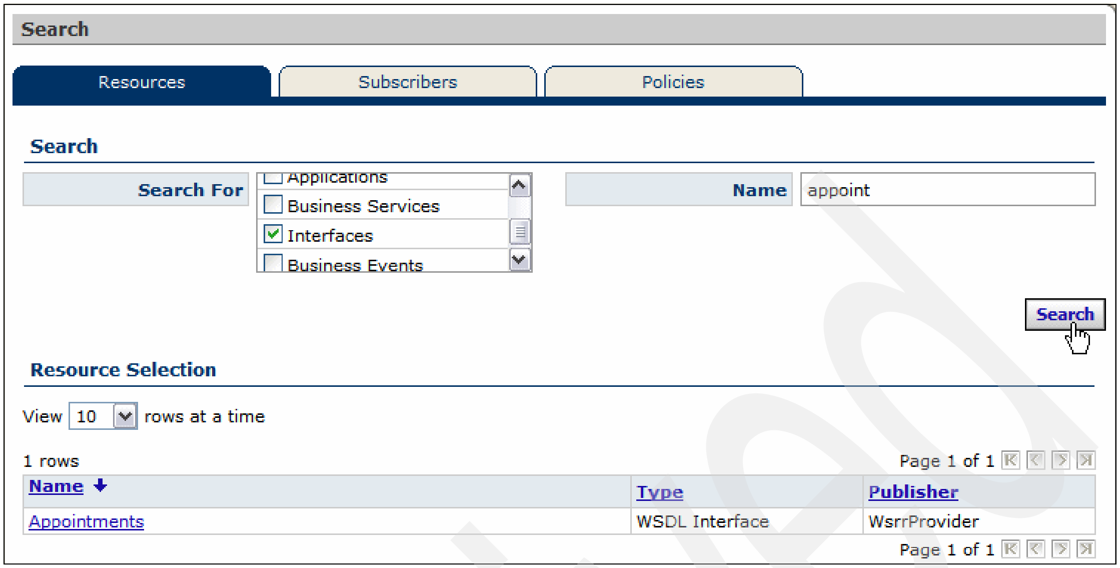


Figure 7-14 Search Repository in WebSphere Business Services Fabric

7.4 Consuming business services from processes

This section contains these topics:

- Dynamic assembly of services
- WebSphere Business Services Fabric objects
- Dynamic Assembly SCA component

Dynamic assembly of services

WebSphere Business Services Fabric introduces dynamic capabilities that can be added to WebSphere Process Server using Business Process Execution Language (WS-BPEL) business processes running in a WebSphere Process Server environment. This is done through the Business Services Dynamic Assembler component provided with WebSphere Business Services Fabric, as illustrated in Figure 7-15 on page 210.

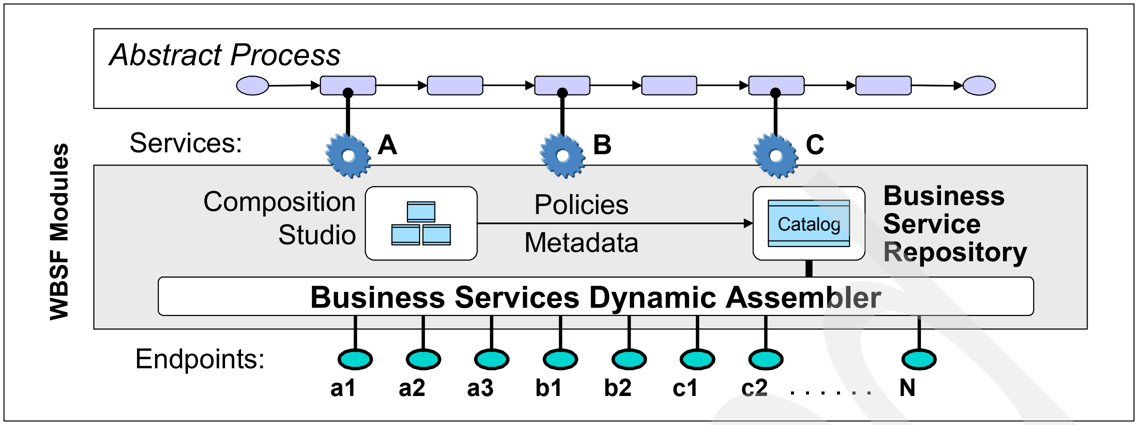


Figure 7-15 Abstract process and how dynamic capabilities are implemented

In our scenario, we have defined our business process, and, in WebSphere Integration Developer, our abstract process instantiated as WS-BPEL looks like Figure 7-16 on page 211.

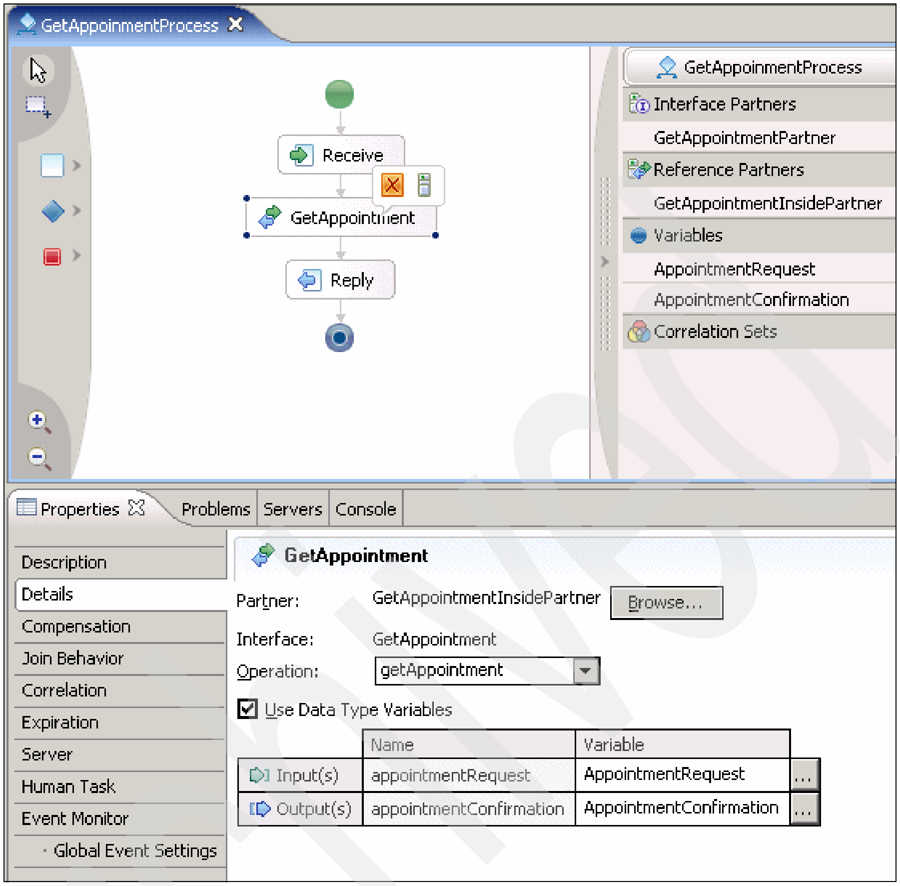


Figure 7-16 BPEL process design in an abstract way

WebSphere Business Services Fabric provides a runtime environment where processes run and use new components that allow the process to semantically decide which service must be requested.

After services are invoked inside a WebSphere Business Services Fabric environment, the life cycle of a service invocation can be described as in Figure 7-17 on page 212, and this is the key to understanding how WebSphere Business Services Fabric works.

In our scenario, consumers will invoke a WS-BPEL exposed interface to instantiate our business process. When the process invokes the `GetAppointment` activity (we only provide one business service), WebSphere Business Services Fabric will determine which endpoint (external or internal provider) is the most suitable.

We will later explain how we build the policy for this request, but in Figure 7-17, we identify our final service providers (internal and external) as the `Endpoint` component.

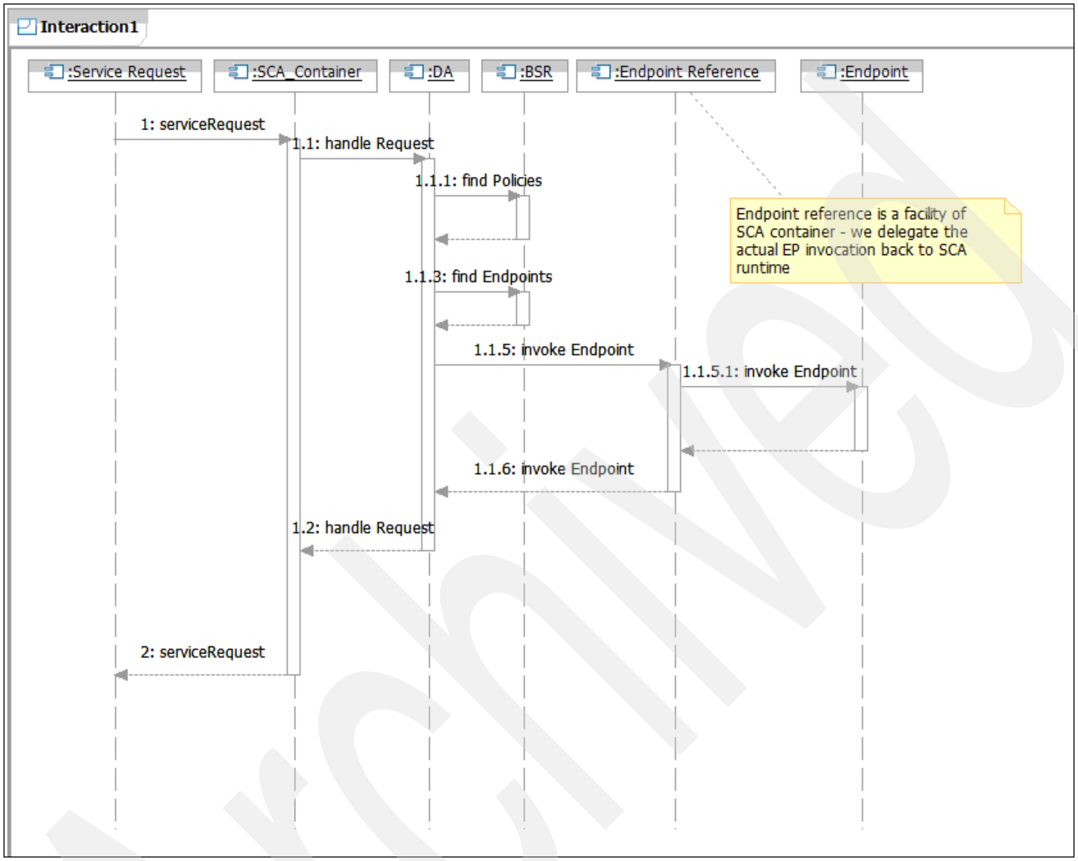


Figure 7-17 Service invocation life cycle at WebSphere Business Services Fabric

First, a context must be created. The *context* is necessary for WebSphere Business Services Fabric, because it stores information that it will use for many things when invocation occurs, such as users, subscribers, and interfaces that can be used for policy assertions. Context identification is part of each SOAP message. This is how WebSphere Business Services Fabric works:

1. Create subcontext

   First, context and content information is used to create subcontexts, which are later used to update the context itself.

2. Build policies

   From the updated context and policy definition, the policy is built in order to apply it to a list of candidates.

3. Find candidates

   From the list of all possible candidates (endpoints), it applies policies in order to filter them and get the most suitable endpoint to call. Each endpoint will also provide an *assertion*, which is used by WebSphere Business Services Fabric to classify if the endpoint is suitable or not.

4. Invoke

   Invoke the service at the selected endpoint.

5. Response

   If no error is produced, then the response is produced.

6. Manage errors

   Resolve errors related to the ability to find suitable candidates or errors relating to the invocation of services themselves.

WebSphere Business Services Fabric objects

WebSphere Business Services Fabric has its own core ontology model in order to facilitate model-driven architecture building. We have previously defined a project in WebSphere Business Services Fabric that will store everything related to our scenario, so we use this project as defined in 7.2.3, "Synchronizing with WebSphere Integration Developer" on page 200.

Therefore, we must define how our scenario will be represented in WebSphere Business Services Fabric:

- *Business service*

  This is a specific business function whose behavior can change during run time for some factors, such as the operating context of the request, policies, and the request content. There are three types of business process: *Process Service, Visibility services*, and *Optimization services*. We will implement a Process Service with the name `ITSO-Tel ManageWorkforce PS` as shown in Figure 7-18 on page 214.

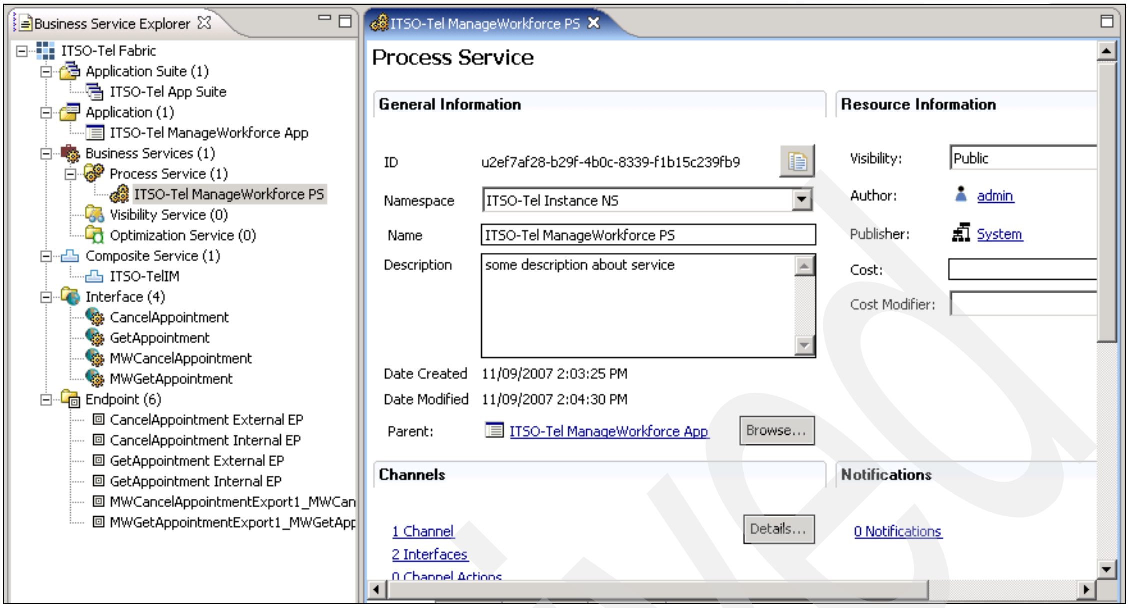


Figure 7-18 Business Services view from WebSphere Integration Developer

- Applications and application suites

  An *application* is a collection of business services in order to solve a business requirement. This is also known as a *composite business service*. An *application suite* is a collection of applications. In our scenario, we have defined `ITSO-Tel App Suite` and `ITSO-Tel ManageWorkforce App` as one application of our suite, which will contain the previous business service, as shown in Figure 7-19 on page 215.

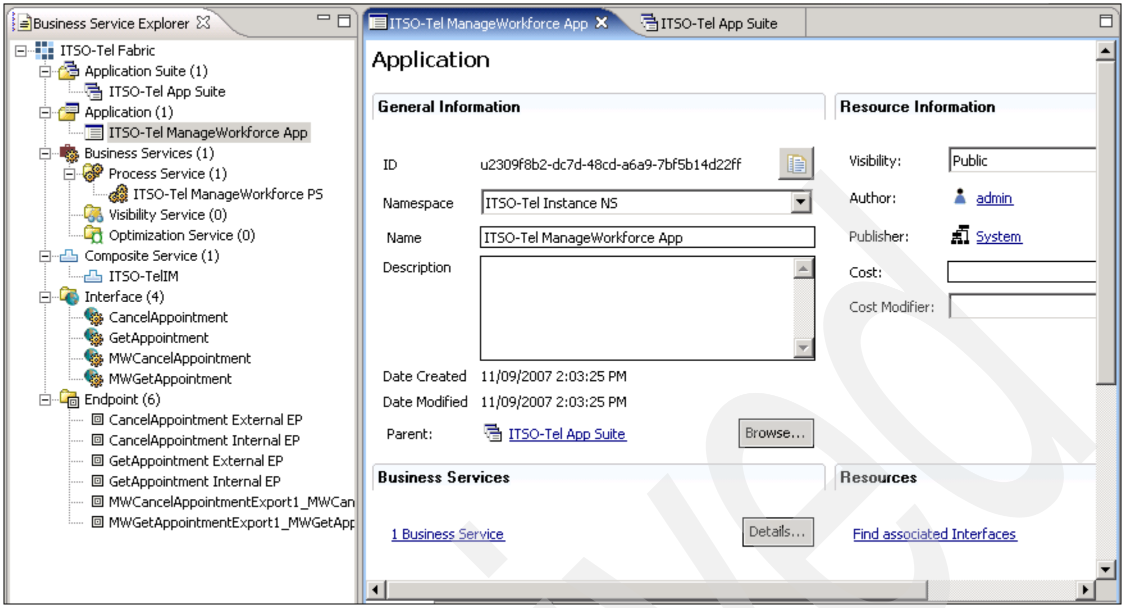


Figure 7-19 Application view from WebSphere Integration Developer

7.4.1 Dynamic Assembly SCA component

Service Component Architecture (SCA) lets us create composite services from modular software components. Using WebSphere Integration Developer to develop SCA components allows us to specify which services will be used through dynamic assembly, as shown in Figure 7-20 on page 216.

Figure 7-20 on page 216 shows how consumers of the `CancelAppointment` and `GetAppointment` interfaces send requests to WS-BPEL business processes that in turn use dynamic assembly components to determine which endpoint is to be used.

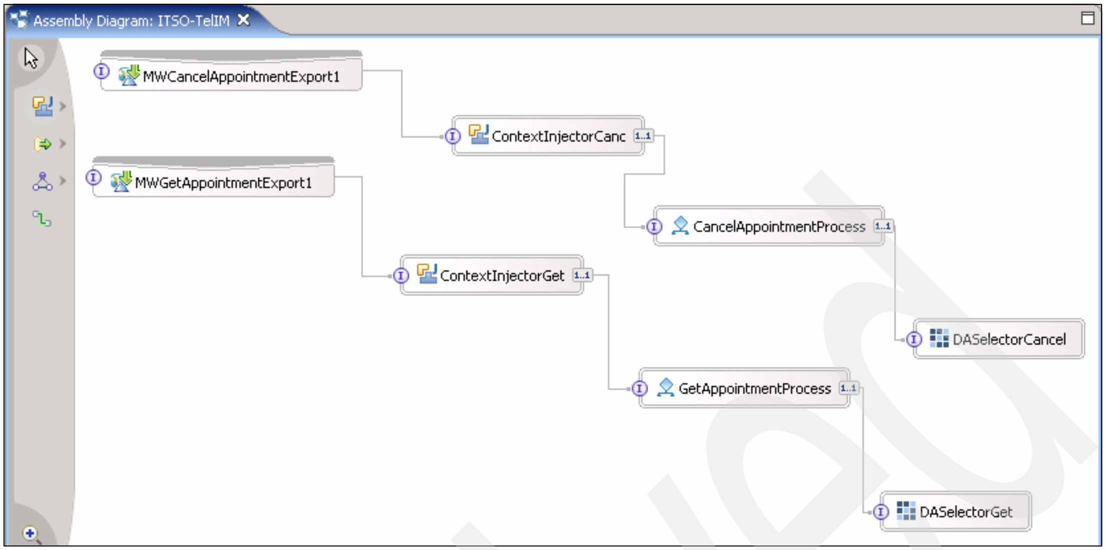


Figure 7-20 Dynamic assembly SCA implementation

Assigning endpoints to interfaces

Interfaces can define multiple endpoints, each of which is a candidate to be selected at run time. Endpoints are associated to one or more interfaces.

In our scenario, we have defined the interface GetAppointment. This has two endpoints defined for it, as shown in Figure 7-21 on page 217.

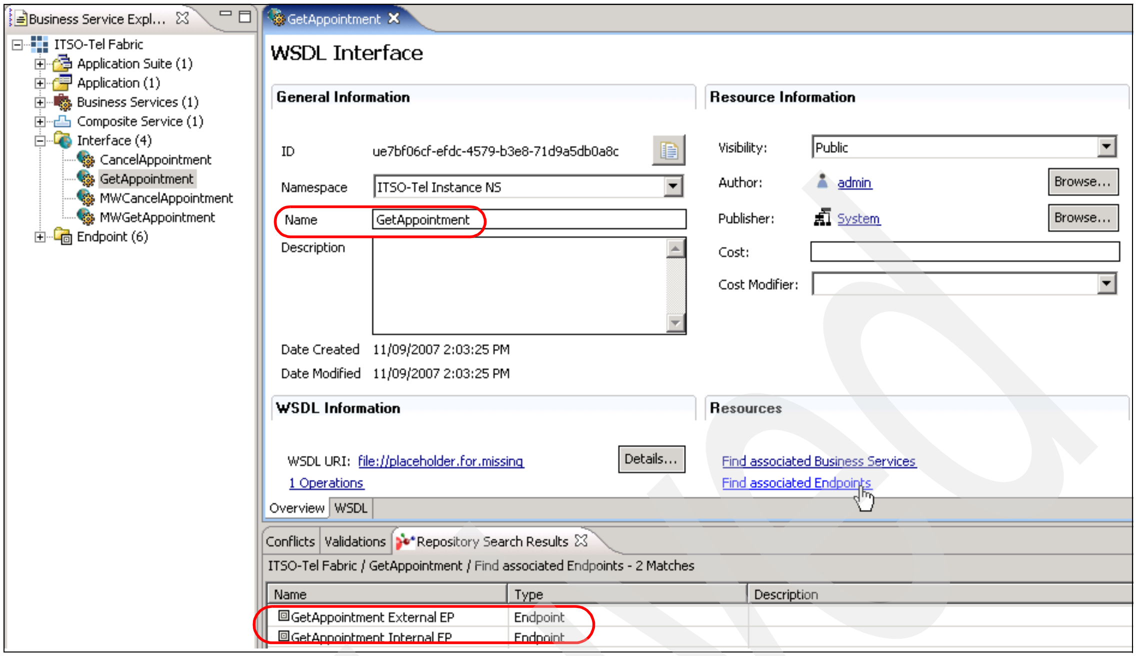


Figure 7-21 GetAppointment interface view in WebSphere Integration Developer

These two endpoints have been defined, as shown in Figure 7-22.

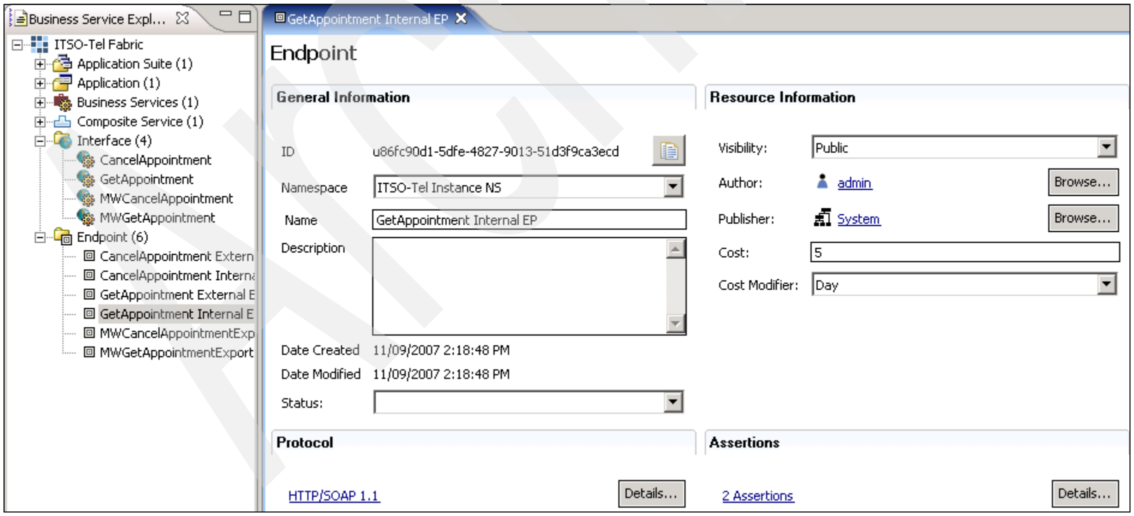


Figure 7-22 GetAppointment Internal EP endpoint view in WebSphere Integration Developer

Endpoint assertions

After you define endpoints, you must specify assertions in order to determine alternatives in which each endpoint is eligible. Assertions in the WebSphere Business Services Fabric Foundation pack installation are classified into five groups, as shown in Table 7-1.

Table 7-1 Endpoint assertions

Assertion group	Type of assertion
Performance	Availability, Maximum Transaction time, or Minimum success rate
Reliability	Delay between retries, Delivery ratio, Delivery Retry, or Timeout
Interoperability	Data format, Delivery channel, Email notification, Fax notification, Inbox notification, Supports version, Transformation, Units of Works, or WSI Profile
Security	Channel Action Grant, HTTP Basic Auth, HTTP Digest Auth, HTTPS Client Auth, Kerberos Token, SAML, SSL, Username Token, X509 Certificate Token, XACML, XML Digital Signature, XML Encryption, or XRML License
Manageability	Deprecation, Hours of Operation, or Propagate Policy

Assertions are specified for endpoints as shown in Figure 7-23 on page 219.

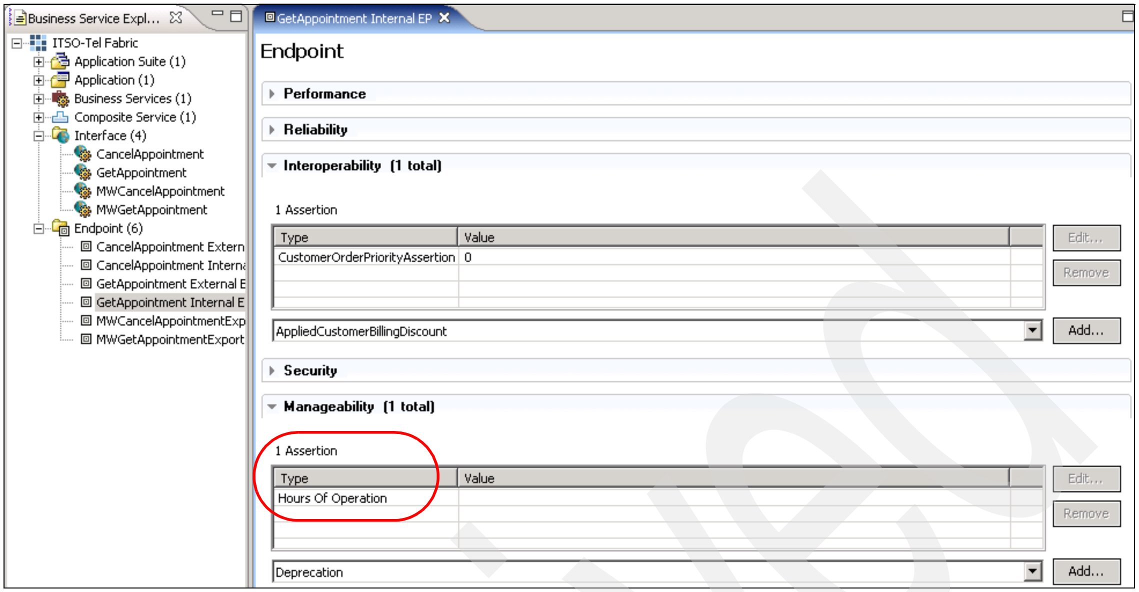


Figure 7-23 Assertion view for endpoints in WebSphere Integration Developer

In our scenario, we will define two assertions: one assertion for CustomerOrderPriorityAssertion and another assertion for Hours Of Operation.

When more than one endpoint has eligible assertions, there is another property that helps WebSphere Business Services Fabric to choose the best endpoint. That property is *Cost*. This value can determine which is the cheapest endpoint among all candidates.

Policies

Assertions allow WebSphere Business Services Fabric to make decisions about which endpoint must be used. However, policies are a rich tool to tell WebSphere Business Services Fabric which services are the best to use.

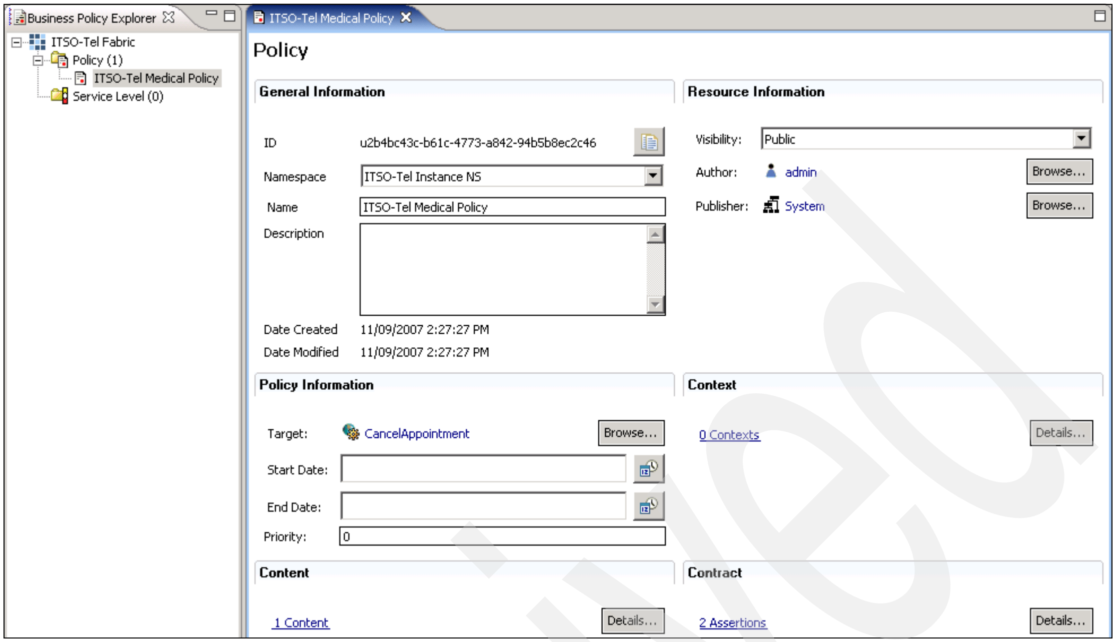


Figure 7-24 Business Policy view in WebSphere Integration Developer

Policies are evaluated in this way:

```
if CONDITION then call endpoint which assertion match ASSERTION
```

Where `CONDITION` is built with `Context` and `Content` information, and `ASSERTION` is built with `Contract`. Therefore, WebSphere Business Services Fabric will try to find the most suitable endpoint to solve the requirement.

In our scenario, we need a policy to identify when the customer requires medical priority. As described in Table 4-4 on page 100, there is an SLA from the `Medical Priority` category defined `SLA-01` that we must achieve. We will define a business policy for this:

- `CONDITION` - Content has `assignedPriority` of 1
- `ASSERTION` - Contract has `CustomerOrderPriorityAssertion` of 1, `Hours of Operation` of 24 x 7.

Our policy is built reusing some Telecom Operation Package properties, such as `CustomerOrderPriorityAssertion` (representing which kind of customer it is for: `0` for non-medical priority customers and `1` for medical priority customers).

To define this policy, perform these steps:

1. Click the **Business Policy Explorer** tab. Right-click the **ITSO-Tel Fabric** folder and select **New** → **Policy**.
2. Enter details for the policy, and then click **Next** (Figure 7-25):
   a. Project: **ITSO-Tel Fabric**
   b. Name: `ITSO-Tel Medical Policy`
   c. Namespace: **ITSO-Tel Instance NS**

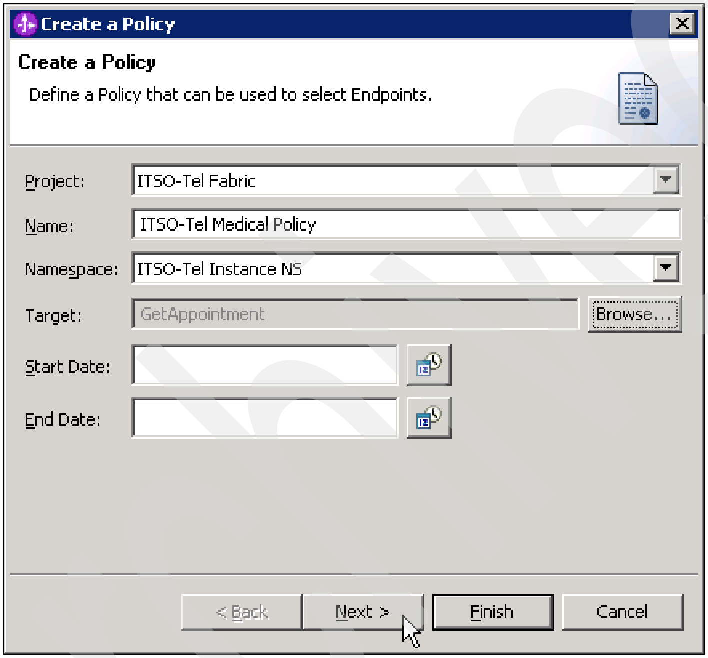


Figure 7-25 Creating a policy

3. Select the content conditions to apply to the policy, and then click **Finish** (Figure 7-26 on page 222).

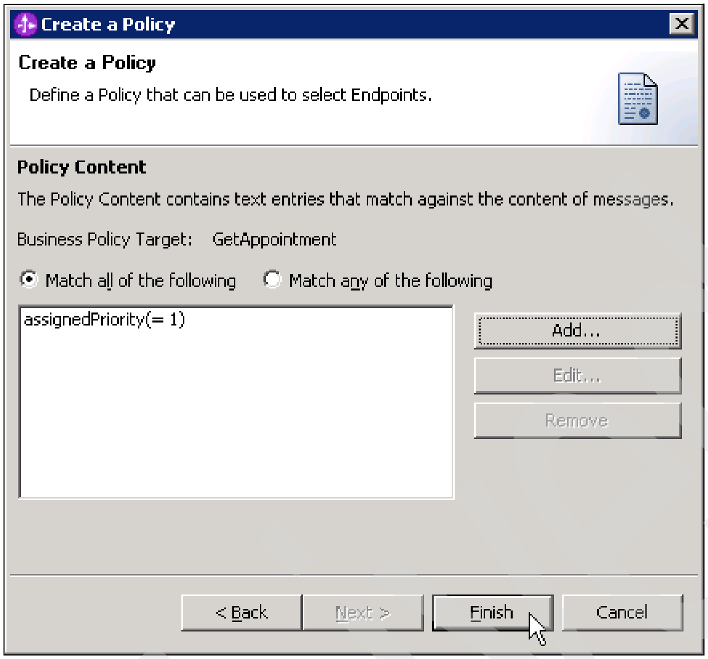


Figure 7-26 Content conditions for the policy

4. Finally, specify the assertions for the policy, as shown in Figure 7-27.

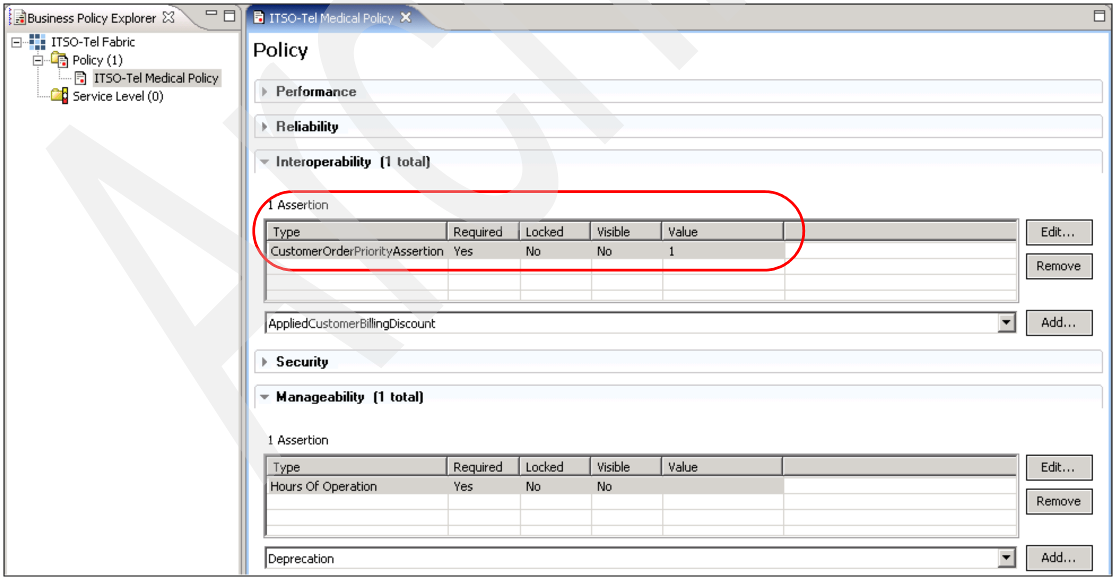


Figure 7-27 Assertion specification for the policy

You can use a similar approach to define a policy during the AFL finals in the state of Victoria with start date of `1 September` through to end date of `30 September`, as previously defined for functional requirements `FR-06` and `FR-07`, described in Table 4-2 on page 97.

Therefore, we define another business policy:

- `CONDITION` - `productName` = "Cable TV", state = "VIC"
- `ASSERTION` - `StateAssertion` = "VIC", `ProductNameAssertion` = "Cable TV"

After you define policies, you can simulate the policies to test their behavior prior to submitting them into the WebSphere Business Services Repository.

Now, that you have defined interfaces, endpoints, and policies, you can see a complete abstract sequence diagram of the `getAppointment` interface being requested inside our business service `ITSO-Tel ManageWorkforce PS`. See Figure 7-28.

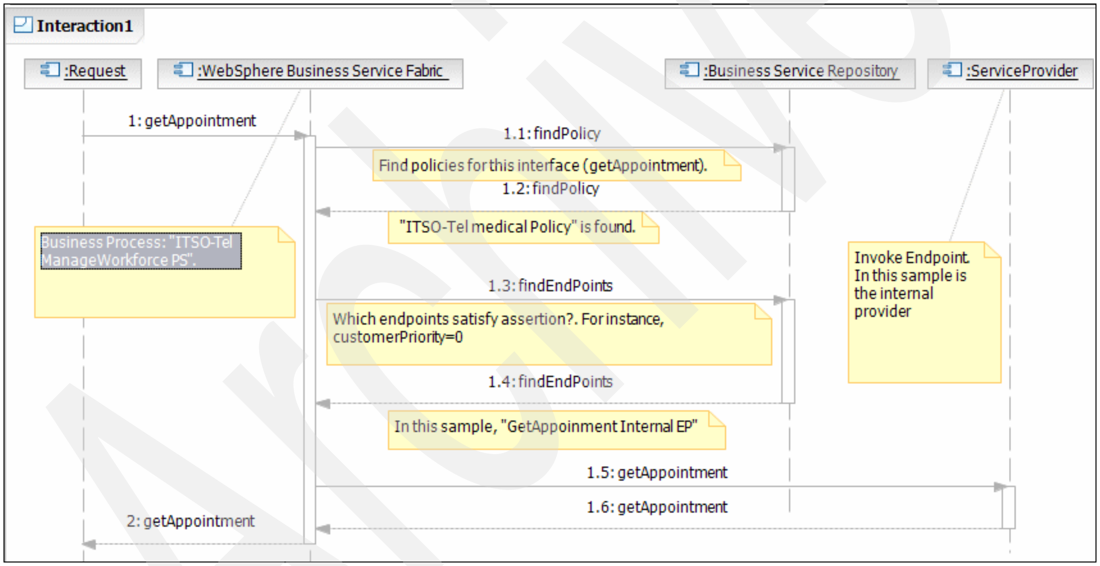


Figure 7-28 Abstract sequence showing a complete execution of a `getAppointment` request

7.5 Managing and governing services

This section discusses:

- Business Services Governance Manager
- Business Services Subscriber Manager

7.5.1 Business Services Governance Manager

The Business Services Governance Manager is the main tool for managing governance in WebSphere Business Services Fabric. You must define projects and namespaces in WebSphere Business Services Fabric using the Governance Manager.

In our scenario, we defined three namespaces in the Governance Manager as shown in Figure 7-29.

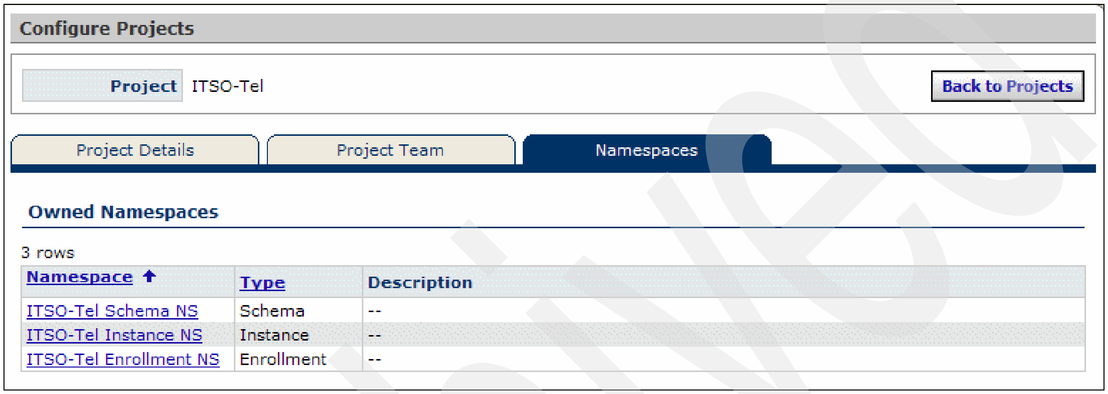


Figure 7-29 Namespaces for our ITSO-Tel project

We use this tool in order to manage:

- Application suite: `ITSO-Tel App Suite`
- Application: `ITSO-Tel ManageWorkforce App`
- Business services:
  - Process service: `ITSO-Tel ManageWorkforce PS`
  - Visibility service
  - Optimization service
- Composite service: `ITSO-TelIM`
- Interfaces:
  - `GetAppointment`
  - `CancelAppointment`
  - `MWGetAppointment`
  - `MWCancelAppointment`

- Endpoints:
  - `CancelAppointment External EP`

    This endpoint represents the external entity that provides the service that cancels appointments. It implements the `CancelAppoinment` interface.
  - `CancelAppointment Internal EP`

    This endpoint represents the internal entity that provides the service that cancels appointments. It also implements the `CancelAppoinment` interface.
  - `GetAppointment External EP`

    This endpoint represents the external entity that provides the service that gets appointments. It implements the `GetAppoinment` interface.
  - `GetAppointment Internal EP`

    This endpoint represents the internal entity that provides the service that gets appointments. It also implements the `GetAppoinment` interface.
  - `MWCancelAppointmentExport1_MWCancelAppoinmentHttp_Service`

    This endpoint represents the exposed service for the `MWCancelAppoinment` interface.
  - `MWGetAppointmentExport1_MWGetAppoinmentHttp_Service`

    This endpoint represents the exposed service for the `MWGetAppoinment` interface.

Any changes made to these entities in WebSphere Integration Developer are replicated with WebSphere Business Services Fabric and reported in the *Changelist* in the Governance Manager, as seen in Figure 7-30 on page 226.

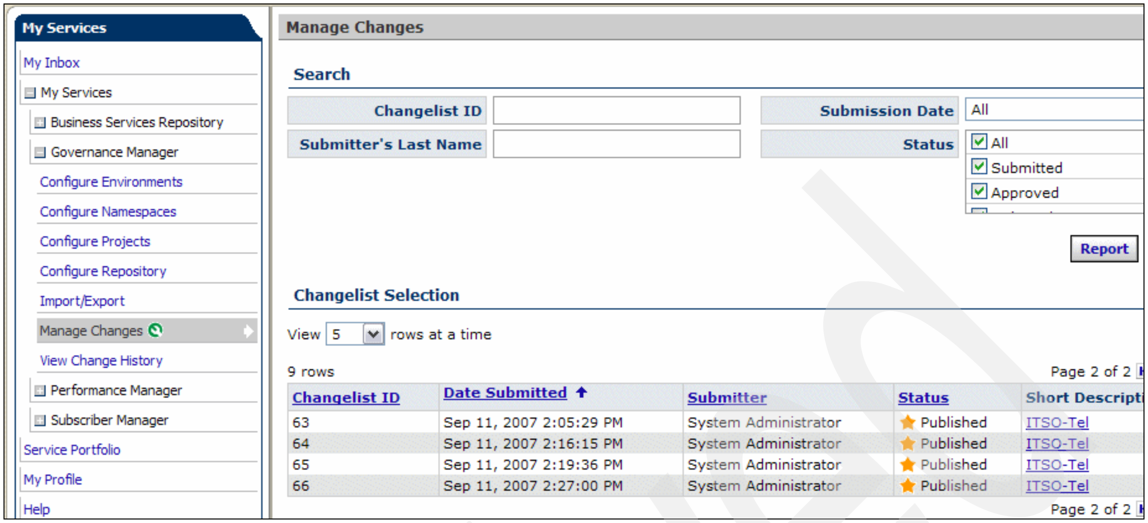


Figure 7-30 Changelist view in the Governance Manager

7.5.2 Business Services Subscriber Manager

The Business Services Subscriber Manager is a Web-based console that manages business service entitlements using an organizational enrollment and subscription model.

All consumers must be subscribers to services in order to control service access and to notify when events occur.

Organizations are enrolled to a service (application or application suite). In our scenario, we enroll the `System` organization to the `ITSO-Tel Application Suite`, `Application`, and `Business Service`. Users who belong to the `System` organization are then subscribed to the `ITSO-Tel service`.

7.6 Measuring services

Access to services is controlled in many ways. WebSphere Service Registry and Repository does not control access to services, because it only provides information about where the services are. WebSphere Service Registry and Repository can control services information access, but not access to the service itself.

However, WebSphere Business Services Fabric provides services infrastructure, so that all services accessed inside the WebSphere Business Services Fabric environment can be controlled and therefore can be measured by it.

7.6.1 Business Services Performance Manager

Business Services Performance Manager is a Web-based console that provides visibility and monitoring of business services-based solutions. This module includes a set of reports that enables administrators to monitor the behavior and performance of their business services. We describe several of the measurements in this section.

Ecosystem performance

This type of reporting provides information about success and failures when invoking services inside WebSphere Business Services Fabric. Reports about certain periods and certain service categories are displayed in graphics and spreadsheets. Several metrics include:

- Success/failure by period
- Success/failure by business service category
- Message success/failure analysis over a certain period

An example of an ecosystem report is shown in Figure 7-31 on page 228.

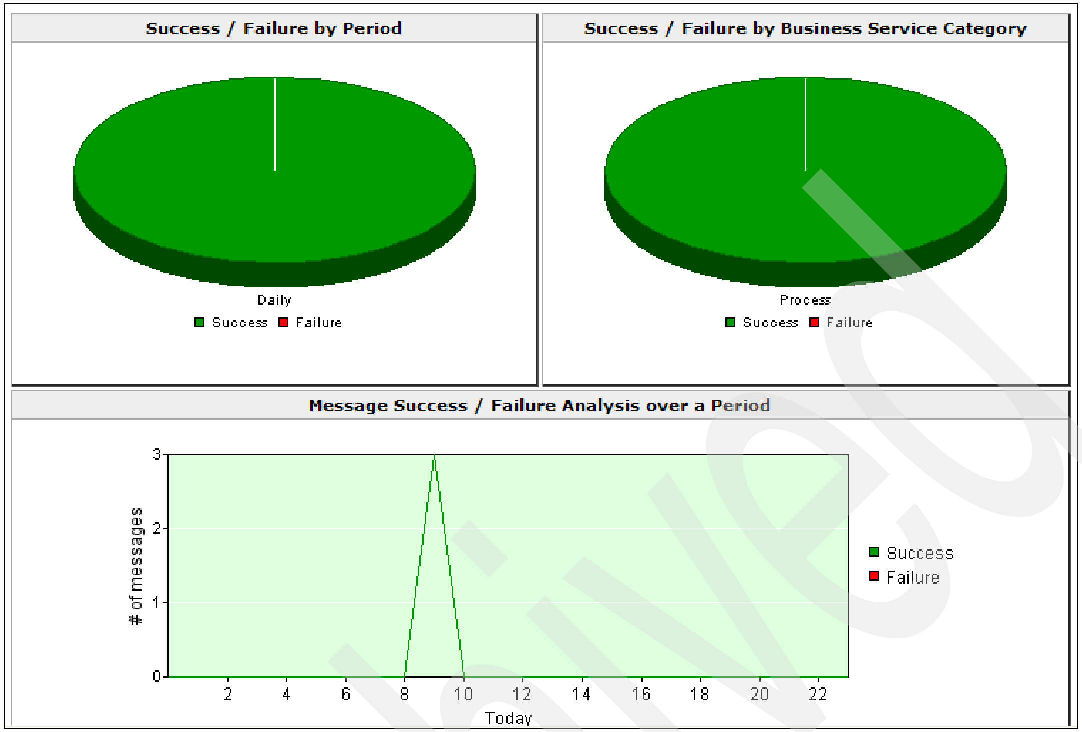


Figure 7-31 Ecosystem report

Service invocation summary

This type of reporting provides information about response time and the channel services used. In our scenario, we have defined many SLAs, and `Response Time` is one of them. Therefore, we can use this report to control service behavior.

Service performance

This type of reporting is relevant to interfaces, as well as endpoints. It provides reports for other SLAs. These reports are:

- Transaction Volume/Response Time

  Shows a metric that compares the volume of a certain transaction against its response time.

- Availability

  Shows how much time (as a percentage) that the service is available.

In our scenario, we can monitor our interfaces and endpoint behavior, as shown in Figure 7-32 on page 229.

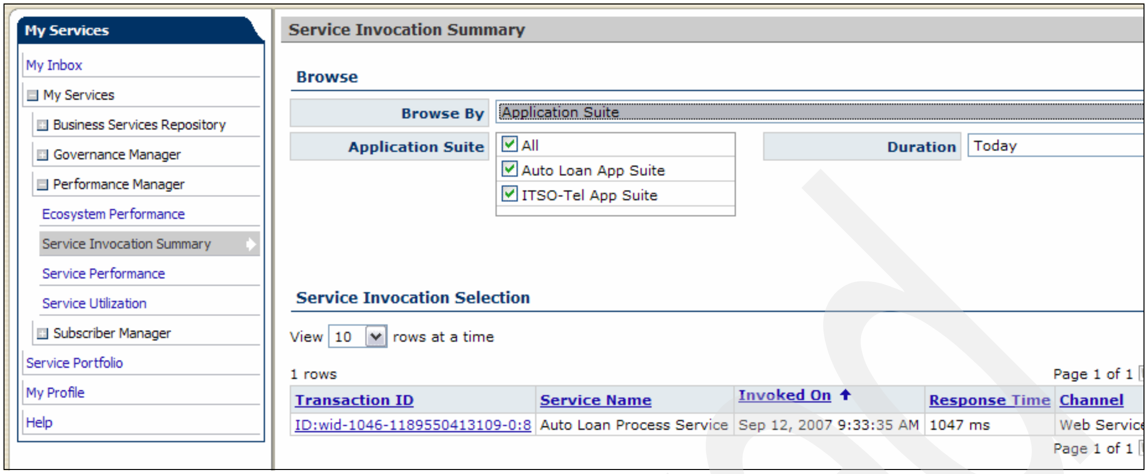


Figure 7-32 Service Invocation summary

Service utilization

This part of the reporting shows how much a service is used. Only business services that do not belong to the System organization are shown. Because all administrative capabilities of WebSphere Business Services Fabric are modeled as business services and assigned to the System organization, they are filtered out. In our scenario, if we want to see them, we must assign our project to another organization.

7.6.2 IBM Tivoli Composite Application Manager communication

WebSphere Service Registry and Repository provides a plug-in to integrate with IBM Tivoli Composite Application Manager. Using this plug-in, notification events can be monitored by Tivoli Composite Application Manager as part of an SOA management solution.

This plug-in is a J2EE application, which will connect to an IBM Tivoli Composite Application Manager port every time that events happen inside WebSphere Service Registry and Repository. Therefore, if you use IBM Tivoli Composite Application Manager for SOA management, you need to use this plug-in for the mechanism to communicate events.

8

Data mediation with Progress DataXtend Semantic Integrator

In this chapter, we discuss layered mediation architecture support for service-oriented architecture (SOA) governance and how to use DataXtend Semantic Integrator to implement service components for data mediation.

This chapter uses the fictional organization ITSO-Tel (as described in Chapter 4, “Our business scenario” on page 87) to illustrate the use of DataXtend Semantic Integrator.

This chapter contains the following sections:

- Layered mediation component model
- Data model governance with DataXtend Semantic Integrator
- Implementing the external workforce adapter

8.1 Layered mediation component model

The *layered mediation component* model is a design pattern for the categorization and development of SOA services and service components as discussed in Chapter 5, "Service identification, specification, and realization" on page 101.

Layered mediation is an any-to-any component pattern that addresses the need to transform the structure and representation of information as it is delivered to varied endpoints. You can relate a layered mediation component model to the components found in SOA as requests move from arbitrary requests to a shared service layer and down to arbitrary endpoints (Figure 8-1).

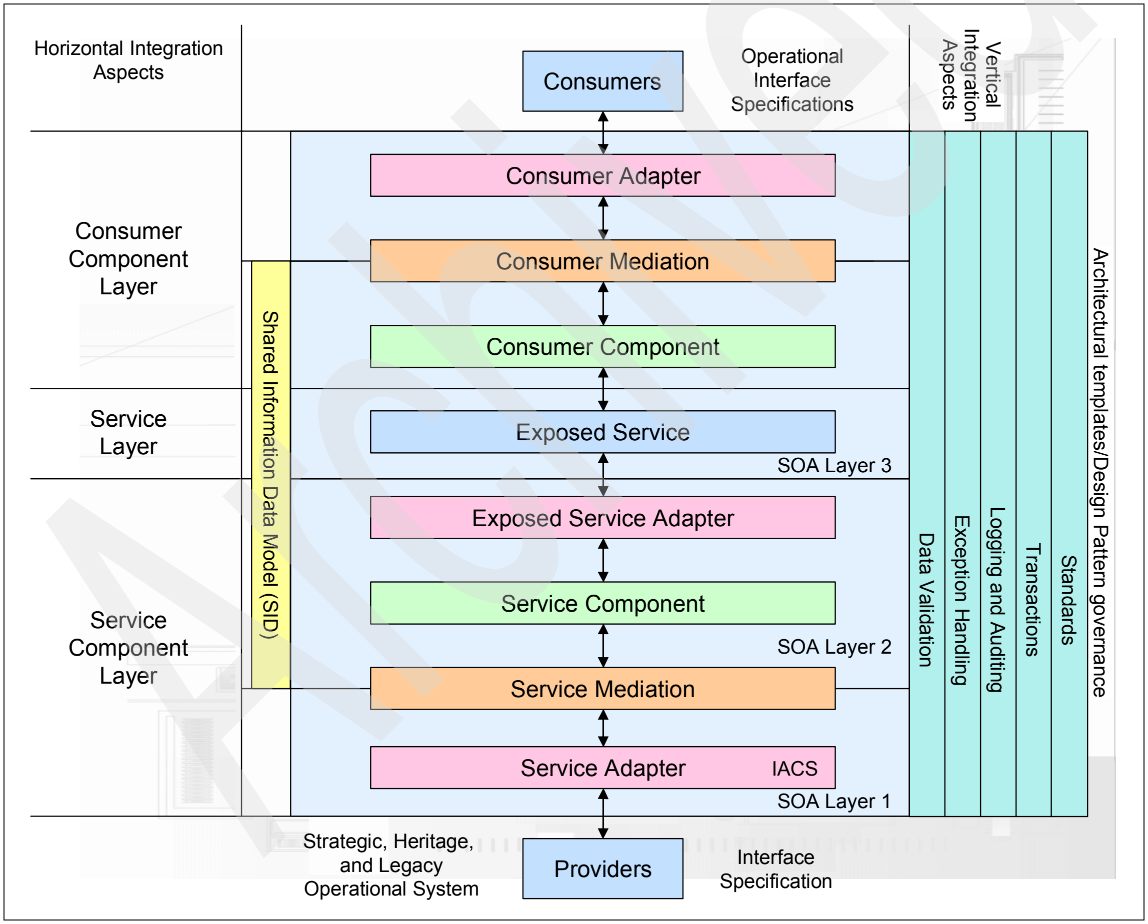


Figure 8-1 Layered mediation component model for SOA

8.1.1 High level horizontal layers

The high level horizontal layers are threefold:

- *Consumer component layer* - The consumer component layer allows exposed services to be consumed irrespective of the calling application format.
- *Service layer* - The service layer is modeled from a telco industry and the ITSO-Tel business perspective.
- *Service component layer* - The service component layer includes all of the software components that are required to realize the service layer.

8.1.2 Low level horizontal layers

Table 8-1 gives you an overview of the low level horizontal layers in the model when applied to the ITSO-Tel SOA architecture.

Table 8-1 Low level horizontal layers

Layer	Responsibility	Governance
Consumers	Inbound Operational system interface	N/A
Consumer adapter	Protocol adaptation - Mapping native transport protocol and wire format to Service Data Objects (SDO)	Support all potential protocol standards? Does the adapter provide data structure validation?
Consumer mediation	Semantic transformation from native data to Shared Information Data (SID)-based common model representation	Is the mapping in a shared Exchange Model to allow reuse and impact analysis?
Consumer component	Logic necessary to call the Exposed Service and provide any needed orchestration	Does the service consume only exposed services? Is the component implemented in Java™ or Business Process Execution Language (BPEL)?

Layer	Responsibility	Governance
Exposed service interface	eTOM/SID-based reusable business service	Does WSDL conform to ITSO-Tel standards, for example, WSDL Basic Profile 1.0 compliant at port type, SOAP message wrapper, or Doc Literal wrapped style?
Exposed service adapter	Protocol adaptation	Does an adapter component exist for all supported protocols, for example, JMS and HTTPS?
Service component	Logic necessary to realize Exposed Service	Is the component implemented in Java or Business Process Execution Language (BPEL)?
Service mediation	Transformation and restructuring of common model data into provider specific message	Is the mapping in a shared Exchange Model to allow reuse and impact analysis?
Service adapter	Mapping from SDO to native wire format and transport protocol	Must not include any service logic or processing
Providers	Outbound Operational system interface	N/A

Additional governance points for all components include requirements that the components must:

- Follow SCA component specifications
- Follow project naming conventions
- Express input and output in SDO format unless communicating with an endpoint system

8.1.3 Low level vertical layers

Table 8-2 gives you an overview of the vertical layers in the model when applied to the ITSO-Tel SOA architecture.

Table 8-2 Low level vertical layers

Layer	Responsibility	Governance
Data validation	To validate all inbound application requests	Validation ensures application does not accept data that it cannot process
Exception handling	To manage all exceptions within the application	Is there a common strategy for exceptions spanning application and component models?
Logging and auditing	To prescribe systems management and the security application aspect	Is there a single logging and auditing strategy application and component model wide?
Standards	To standardize the strategic interfaces	Do all interfaces have traceability to applicable standards?
Transactions	To outline the transaction boundaries	Consistent with system-wide transaction policy?

8.2 Data model governance with DataXtend Semantic Integrator

In our scenario, we discuss the implementation of a new SOA business service where the service component is implemented with WebSphere Business Services Fabric and the Service Adapter is implemented in DataXtend Semantic Integrator.

When seeking to apply data model governance in the SOA, ITSO-Tel chose to implement a Layered Meditation Architecture utilizing DataXtend Semantic Integrator to implement the semantic mediation between components.

In this section, we will discuss the design patterns in DXSI and the use of DXSI to implement the scenario solution.

8.2.1 Common model architecture for data model governance

Common model data architecture is a design constraint to ensure that all information is represented in terms of a single data model shared across a domain or enterprise. Using common model architecture for service design and integration can provide many benefits, including:

- Shared vocabulary in the description and use of services and information
- Reuse of common components
- Alignment with industry and enterprise standards and practices

A common model architecture provides all of the functionality and a higher level of reusability than a data dictionary or common message architecture, because it is better able to capture all interpretations of data and the relationships among types of information across the domain (Figure 8-2).

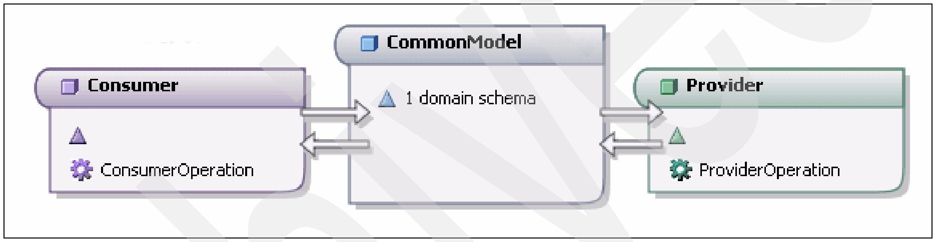


Figure 8-2 Common model architecture

As a next generation service provider, ITSO-Tel bases the common data model on TM Forum standards including eTOM and the SID.

Model-derived services

Common model architecture can also provide guidance and constraint in the design and development of services. Deriving a service whose parameters are based on reusable components of the common model allows for greater reuse and benefits service discovery, specification, and implementation processes (Figure 8-3 on page 237).

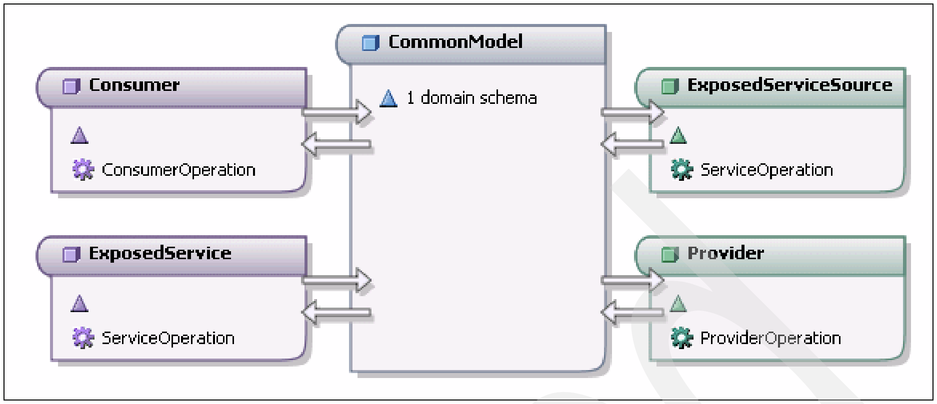


Figure 8-3 Common model architecture with exposed services

In this pattern, the consumption and processing of the consumer message into the exposed service message corresponds to the consumer mediation component in the layered mediation architecture. The processing of the exposed service request into the provider specific message is the service meditation component.

Common model architecture in DXSI

A common model architecture that is implemented in DataXtend Semantic Integrator is called an Exchange Model and is comprised of several elements:

- Data service models - These are inputs to the system.
- Data source models - These are target systems for output or enrichment.
- Common model - This is the shared common model for all information passing through the exchange model.
- Transformation rules - These rules describe the restructuring and translation of objects between the common model and data services or data sources.
- Validation rules - These rules allow for extensive data cleansing and validation.

Impact analysis in a shared exchange model

One of the benefits of capturing the relationships between many models and a common model in a single exchange model is the opportunity for impact analysis. For a particular element in any model, all objects, operations, and services that might be affected by a change can be quickly identified.

8.3 Implementing the external workforce adapter

In this section, we will discuss some of the details of DXSI usage in development of the ITSO-Tel external workforce service mediation component.

8.3.1 Installing DataXtend Semantic Integrator 8.2.1

DataXtend Semantic Integrator (DXSI) is available from Progress Software. In this book, we use Version 8.2.1.

A version bundled with the TM Forum NGOSS SID 7.0 model is known as the *SID Model Browser*. The SID Model Browser is a free download available at:

http://www.progress.com/sid-model-browser

The SID Model Browser is also included in the *Getting Started with the SID* book by John Reilly of the TM Forum. The book is published by TM Forum and is available at:

http://www.tmforum.org/Bookstore/GettingStartedWith/33446/article.html

You can upgrade the SID Model Browser to a full or evaluation version of DXSI by applying an updated license. All installation and usage instructions for DXSI are applicable to the SID Model Browser as well.

You execute a standard executable installer to install DXSI. If there are available fixes or updates, they come with a readme file that indicates how to apply them.

After installing DXSI, note the:

- Memory usage
- File path limitations
- Workspace preferences

Memory usage

When working with large and complex models, you can often improve performance by increasing the memory allocated to the SID model browser. If your machine has more than 1 GB of RAM, we recommend that you increase the maximum allocated memory by modifying <*DXSI Install*>\eclipse\eclipse.ini in your install directory to have a larger value for the -Xmx parameter (Figure 8-4 on page 239).

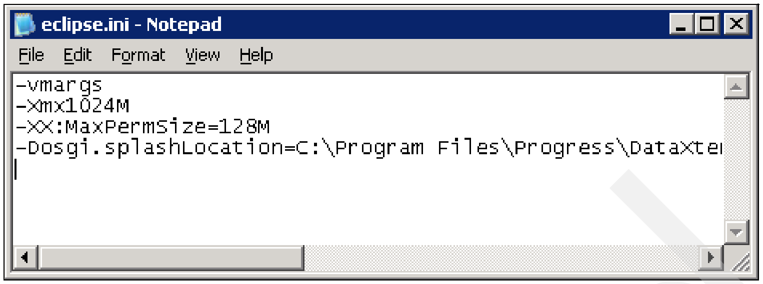


Figure 8-4 Increasing memory allocation

File path limitations

DXSI implements Uniform Modelling Language (UML) and XSD/WSDL-based models as Java classes for runtime processing. Because of the length of UML object names and relationships, XSD/WSDL name spaces, and conventions for Java class building, path names can exceed the maximum length supported by Windows®. We can decrease the likelihood of this situation by minimizing the base path to our workspace and temp directory.

To modify the default Windows temporary directory names, right-click **My Computer** on the Windows desktop and choose **Properties**. Under the **Advanced** tab, choose **Environment Variables** and change TEMP and TMP to use a shorter path, such as C:\temp (Figure 8-5 on page 240).

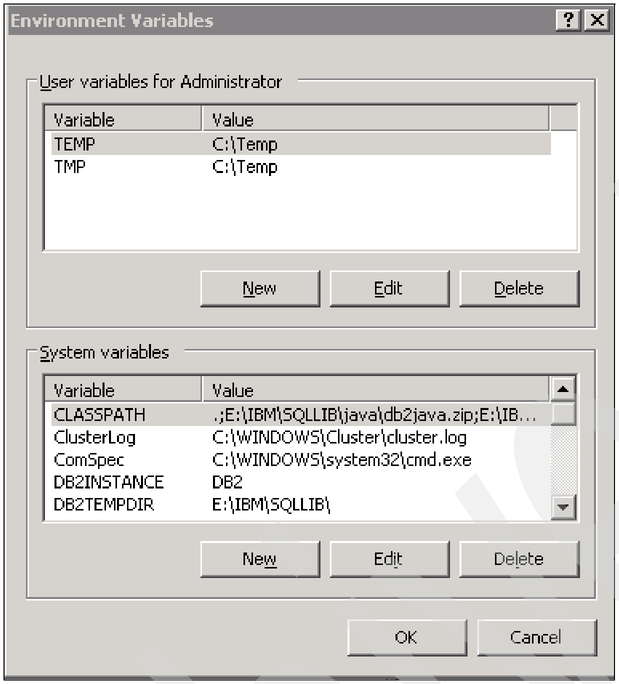


Figure 8-5 Changing TEMP directory

The workspace path is set when you run DXSI for the first time, or you can change it by selecting **File** → **Change Workspace** inside the workbench (Figure 8-6).

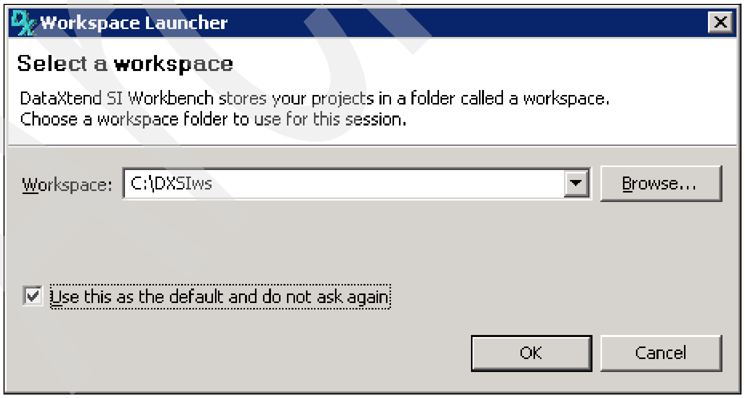


Figure 8-6 Choosing a workspace

Workspace preferences

After you are inside a workspace, turn off Automatic Builds to avoid lengthy recompilation while working in DXSI by deselecting it from the **Project** → **Build Automatically** menu option. If automatic builds need to be disabled, you will see a check mark next to the menu item (Figure 8-7).

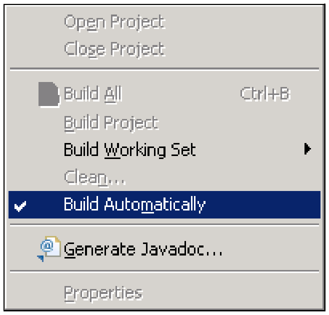


Figure 8-7 Build Automatically menu item prior to disabling

Finally, if you intend to deploy your project to a Java V1.4 runtime environment such as the environment that is used in WebSphere Process Server, you need to change the target Java compliance from the **Window** → **Preferences** → **Java** → **Compiler** settings (Figure 8-8 on page 242).

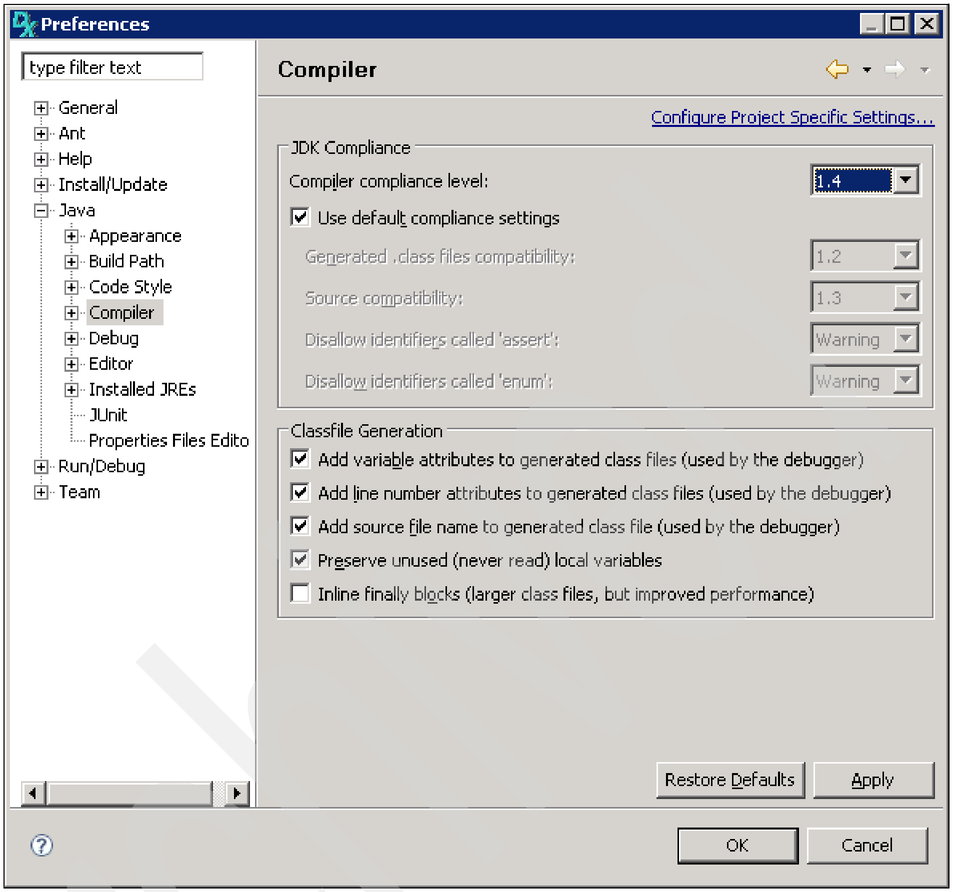


Figure 8-8 Changing target Java compliance

You can also address the Java compliance by running the DXSI workbench under a Java V1.4 Java Runtime Environment (JRE™). Currently, Sun™ Java™ V1.4.2_12 is a supported JRE and you can use it by simply replacing the <*DXSI Install*>\jre directory with the jre subdirectory of the Java V1.4 JDK™, SDK, or JRE.

8.3.2 Importing models

The first step in the creation of an exchange model is the creation or import of models. For ITSO-Tel, we use the SID as the basis of a common model, the preexisting Manage Workforce interface as the data service that we want to realize, and the Manage External Workforce interface as the data source to which we will connect.

Importing the common model

To quickly start our scenario, we use the NGOSS SID directly without any ITSO-Tel extensions as the basis of our common model. You can import the SID directly from the TMF published model in XMI format with some caveats:

- Certain SID attributes are untyped.
- Certain SID class relationships have identical names.
- Certain SID classes have attributes whose type conflicts with parent classes.
- Certain SID classes are abstract and have no concrete subclasses.

These issues will be flagged by DXSI on import and can be easily resolved. However, we can avoid this work by importing the SID from the SID Model Browser project, which has an added benefit that the packages have been color-coded in accordance with SID documentation:

1. To import the model into an ITSO-Tel exchange model from the SID Model Browser project, we choose **DataXtend** → **Import** → **Exchange Model** from the menu. Select **Use a Directory** and browse to select the SID project from your <*DXSI Install*>/samples directory and click **Next**.
2. After the model to import is analyzed, the objects that differ will be displayed. Clear the **Add All** check box and select only the TMForumSIDPhaseVII schema (Figure 8-9). Click **Next** and **Finish**.

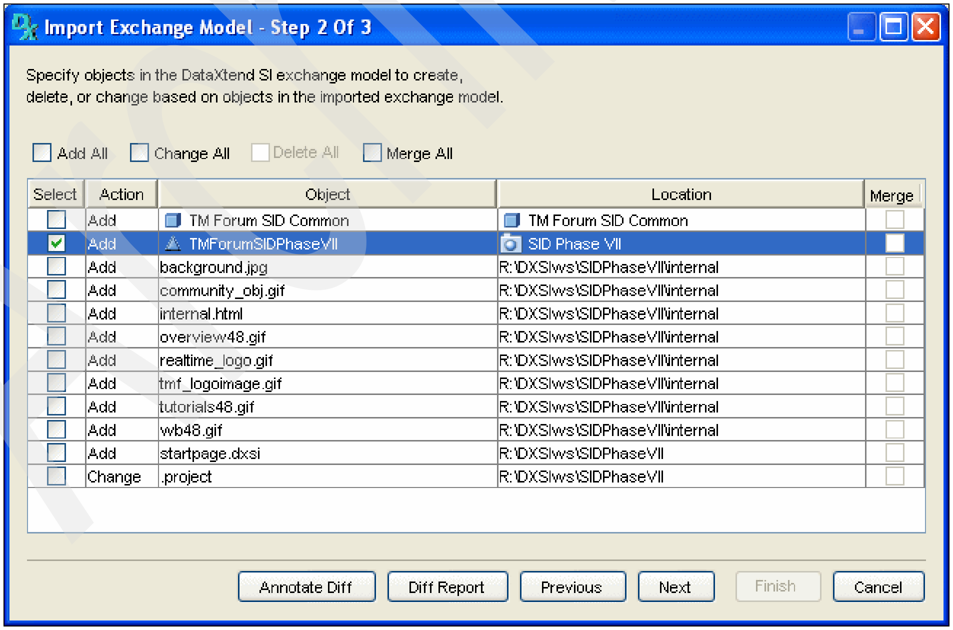


Figure 8-9 Importing just the SID Schema

3. Now, we can create our common model by right-clicking the empty project window and selecting **New** → **New Common Model** (Figure 8-10) from the menu.

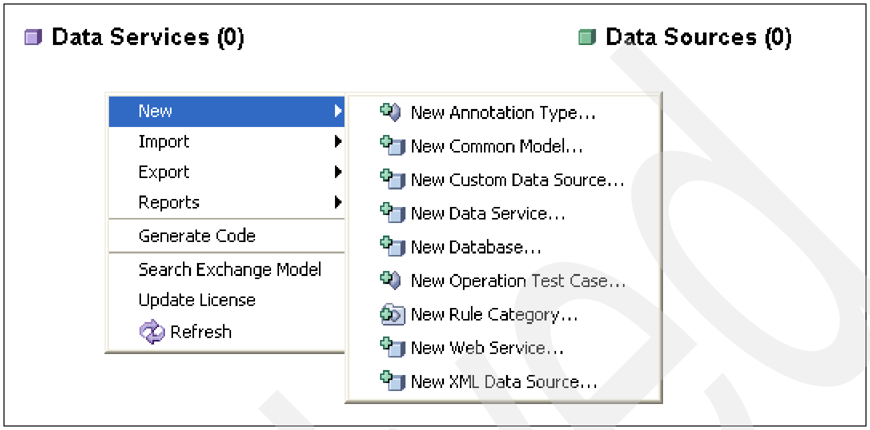


Figure 8-10 Creating a new common model

4. We will name our common model `ITSO_Common`, put it in the Java package `com.itso.cm`, and click to select `TMForumSIDPhaseVII` as our schema.

Building context models

While the SID contains the structures that we need to represent data, our common model also needs to store context for the eTOM operations. To store context for the eTOM operations, we create a package with context classes for the Manage Workforce operation:

1. To create the package in the common model, right-click **ITSO_Common** in the main view pane and select **New** → **Package**. Name the package `OperationalContext` and give it the Java package `com.itso.oc` (Figure 8-11 on page 245).

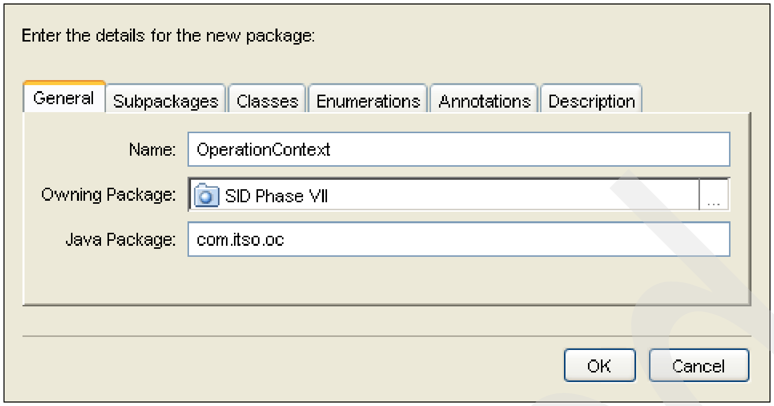


Figure 8-11   Creating the OperationContext package

2. Now, you can find the new package under TMForumSIDPhaseVII under Schemas in the DataXtend Structure view. Right-click to add new classes to represent the specific request and response for the operations. Because our ManageWorkforce service is SID-based, we can see the relevant SID objects to include in the request and response objects quickly.

Importing the WSDLs

To begin integrating our interfaces via the common model, we first need to bring the interfaces into the project. By placing the WSDL and XSD files under the Schemas folder, we indicate that we want these files to be made available in the project:

1. You can import the files either by choosing **File** → **Import**, by right-clicking the **Schemas** folder in the Package Explorer or Navigator, or by simply dragging the files or folders onto the Schemas folder in the Package Explorer or Navigator. If you copy the files to the disk folder without using the Eclipse import, you need to refresh so that Eclipse is aware of the change and the files are shown (Figure 8-12 on page 246).

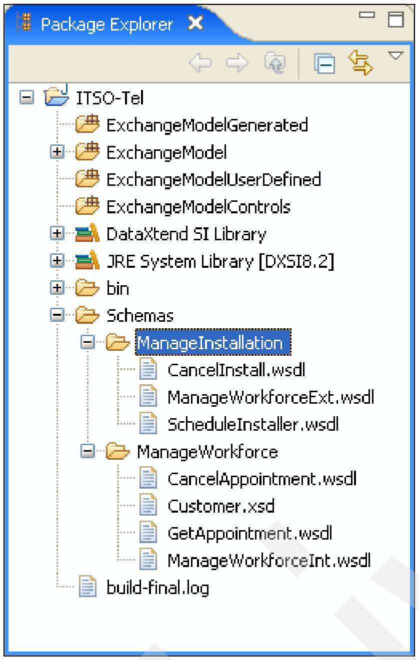


Figure 8-12 Imported interface files in Schemas folder

2. With the files in the project, you can perform a Project Build to confirm that there are no missing files, conflicts, or inconsistencies in the schema structures.

Creating the data service

With the schemas in the project, we can import them into the system as the basis of data sources and data services:

1. Beginning with our internal service, we can import the WSDL as data source by right-clicking the exchange model background or choosing **Import** → **WSDL** from the DataXtend menu. When importing our SID-based internal workforce management interface, we select **Data Service** to create a Data Service and include the Java Package Prefix `com.itso`. Select **Next** (Figure 8-13 on page 247).

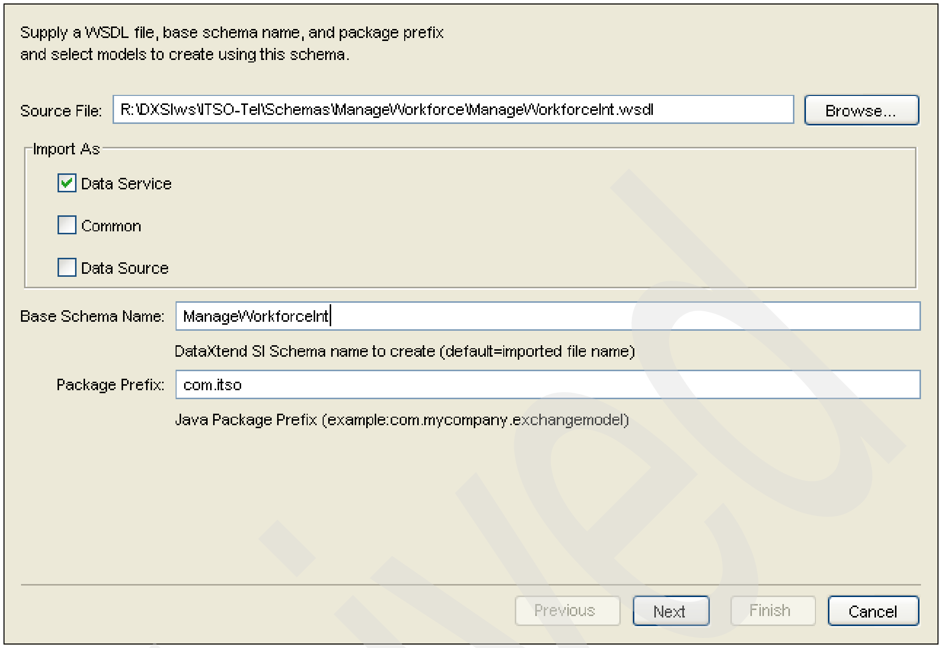


Figure 8-13 Importing the ManageWorkforceInt WSDL

2. When creating the data service, we can shorten the default name to `ManageWorkforce`. Because the service is derived from the common model, we will clear the check box to the left of the Generate Validation Rules option, because that validation can be provided by the identical objects in the common model. We will also select **Generate Maps to Common** and let the tool create the mappings between identically named classes. Select **Next** (Figure 8-14 on page 248).

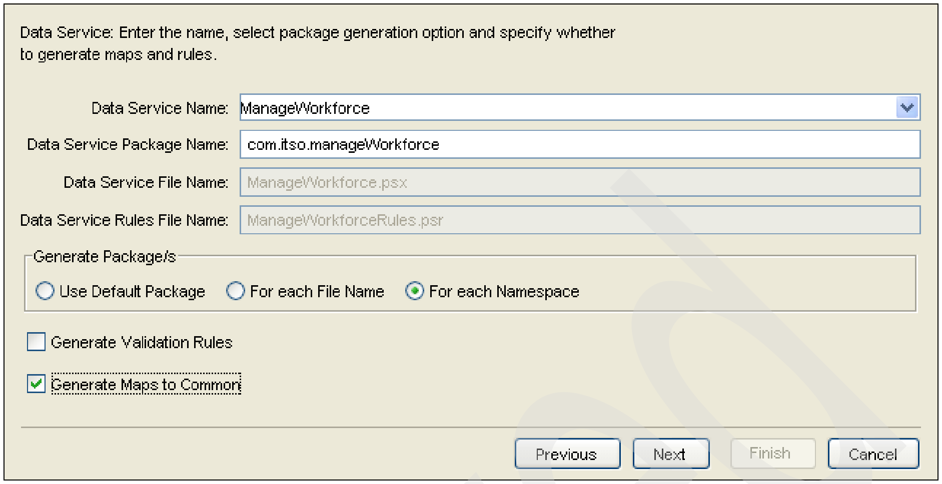


Figure 8-14 Creating the ManageWorkforce Data Service

3. Select **Next** and **Finish** to complete the creation of the data service.

Creating the data source

Similar to creating the data service, we can import the external Manage Workforce WSDL as a data source by right-clicking the exchange model background or choosing **Import** → **WSDL** from the DataXtend menu. When importing this interface, we select **Data Source** to create a Data Source and include the Java Package Prefix `com.itso`. Select **Next**. Because this WSDL is not closely related to the common model, we will accept all the default import behavior by selecting **Next, Next,** and **Finish.**

8.3.3 Establishing the mappings

After the models are imported, you can establish the rules for data translation. For our ITSO-Tel exercise, we make these assumptions to simplify the mapping:

- We have created `CancelAppointmentRequest` and `CancelAppointmentResponse` classes.
- The common model `AppointmentID` for appointments in the external workforce management service is the external system confirmation with an `X` prepended.

To begin the mapping, we double-click the arrow from the common model to the ManageWorkforceWebService (Figure 8-15 on page 249).

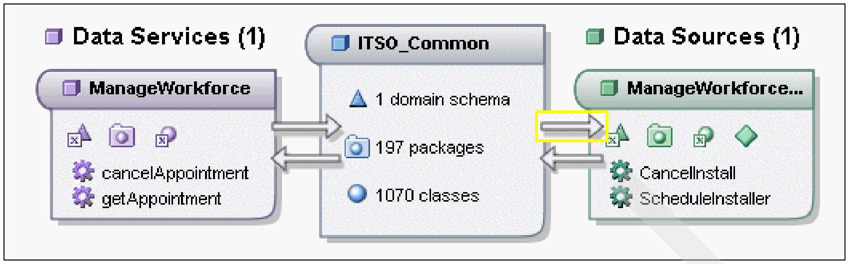


Figure 8-15 Selecting the map to the data source

When we are in the schema map, we drag from our common model schema to the data source schema (Figure 8-16).

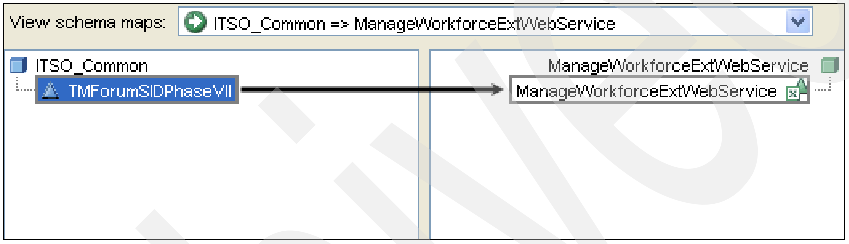


Figure 8-16 Establishing a schema map

After the common model schema is connected to the data source schema, the schema map line is a dashed line to indicate that it is empty. Double-click the dashes to open the schema map, and you will see a list of the class maps. Drag from `CancelAppointmentRequest` to the `CancelInstall` class (Figure 8-17).

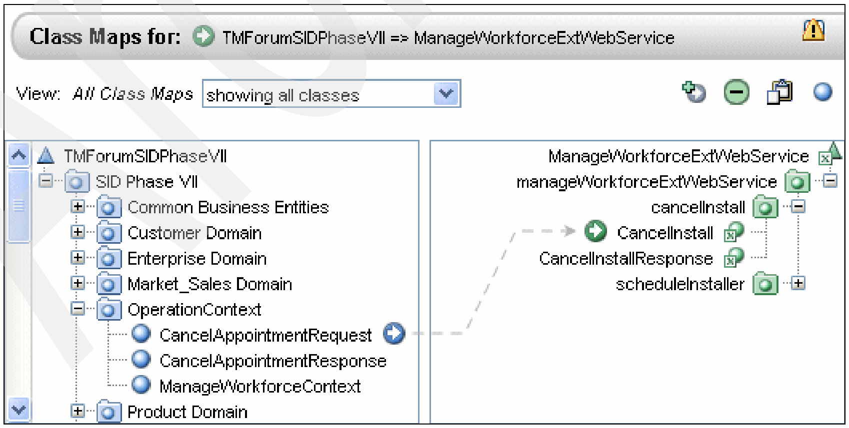


Figure 8-17 Creating a class map

Again, we get a dashed line to indicate that the class map is empty, and we can double-click the map to open it. Here, we can establish the mapping from the `AppointmentID` attribute to the `AppointmentConfirmation` attribute by dragging from one attribute to the other attribute. Initially, this activity defaults to a direct mapping, and we need to modify it in the **DataXtend Editor** → **Source Expression**® tab. By double-clicking the expression line reading `AppointmentID`, we enter the expression builder and can change the source of the mapping (Figure 8-18).

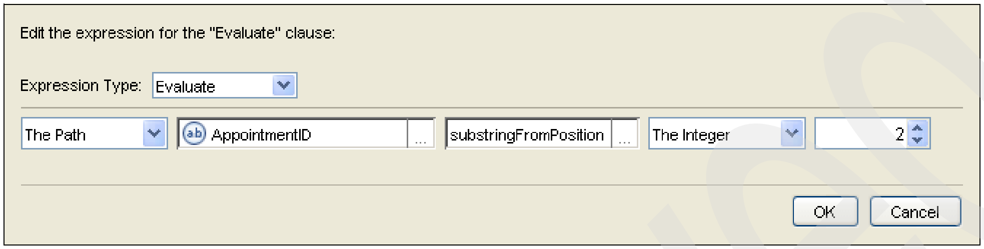


Figure 8-18 The DXSI expression builder

To change the source of the mapping:

1. Choose **The Path** in the first drop-down list box.
2. Choose `AppointmentID` from the path selector.
3. Choose **String Functions** → **substringFromPosition**.
4. Choose **The Integer.**
5. Enter `2` to indicate the second position.

The mapping will then be displayed with an ***fx*** symbol to indicate that it is no longer a one-to-one mapping (Figure 8-19 on page 251).

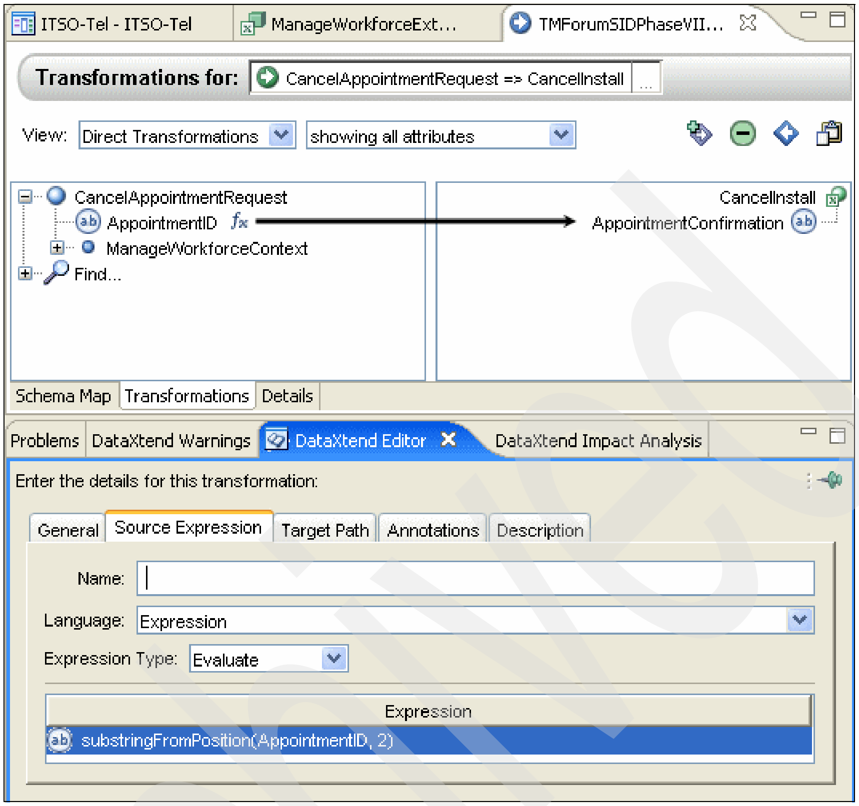


Figure 8-19 Completed AppointmentID mapping

8.3.4 Creating SCA components for WebSphere Process Server

To target the creation of SCA components for use in WebSphere Process Server and WebSphere Business Process Fabric, select the Manage Workforce data service and choose the **DataXtend Editor** → **Code Generators** tab. Click the green (plus symbol) **+** icon to add a new code generator and select **SDO code generator** from the list (Figure 8-20 on page 252).

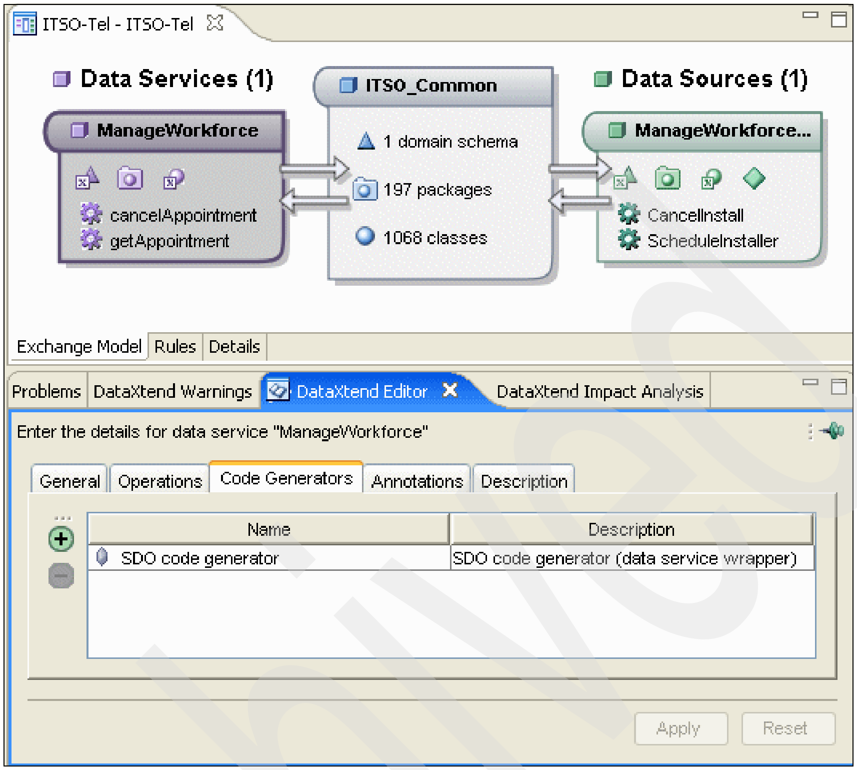


Figure 8-20 SDO code generator

By selecting the SDO code generator, a new folder named `DataObjectServices` will be created at build time and will include source code for any data service with an SDO code generator, in this case, `com\itso\manageWorkforce\ManageWorkforceSDO.java`. You can copy the `DataObjectServices` folder to a WebSphere Integration Developer project to build the SCA around the data service. Or, you can add the folder to the source path of the DXSI project if the library sca.jar from the WebSphere Integration Developer installation is included in the build path.

At run time, it is important that the DXSI run time knows to use SDO as the backing object for messages coming into the data source. While you can set this programmatically at run time, it is usually preferable to change the runtime configuration. To do so, find the runtime configuration in the DataXtend Structure view and set the **XML Source Class** to `SDO XML Entity Source (IBM).` See Figure 8-21 on page 253.

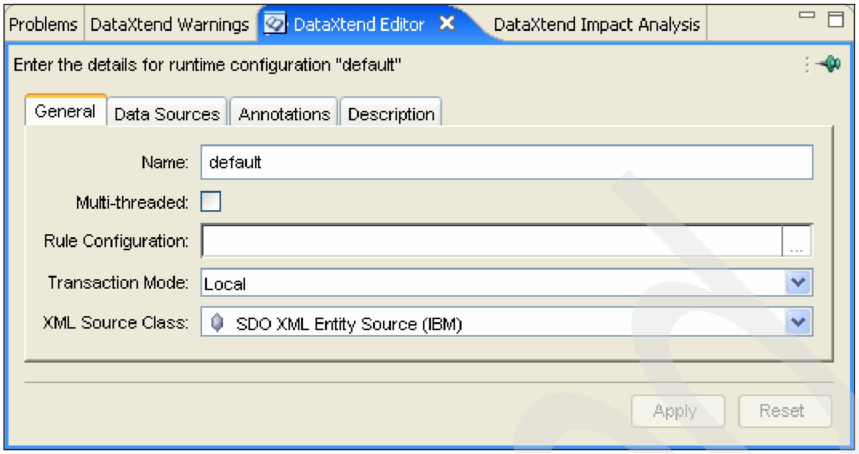


Figure 8-21 Setting the XML source for a runtime configuration

When executing in WebSphere Process Server, note that in addition to the `DataObjectServices` code, you need to include the following project specific libraries and they might change with each build:

- ExchangeModel.jar
- ExchangeModelControls.jar

You can find these libraries in the DXSI project bin subdirectory.

The following required libraries provide DXSI runtime functionality:

- dataxtendsi.jar
- runtimereport.jar
- SDOSource.jar
- backport-util-concurrent.jar
- commons-logging.jar
- log4j.jar

You can find these libraries in the DataXtend SI Library.

Additionally, you can find both sets of jars in DataXtendSI.zip. An archive is created at the end of each build in the project directory.

9

SOA management

This section introduces service-oriented architecture (SOA) management using the following sections:

9.1 Management in the IBM SOA Foundation life cycle

The Manage phase of the IBM SOA Foundation life cycle (see Figure 9-1) includes the tasks, technology, and software that are used to manage and monitor application assets, such as services, business processes, and composite applications that are deployed to the production runtime environment.

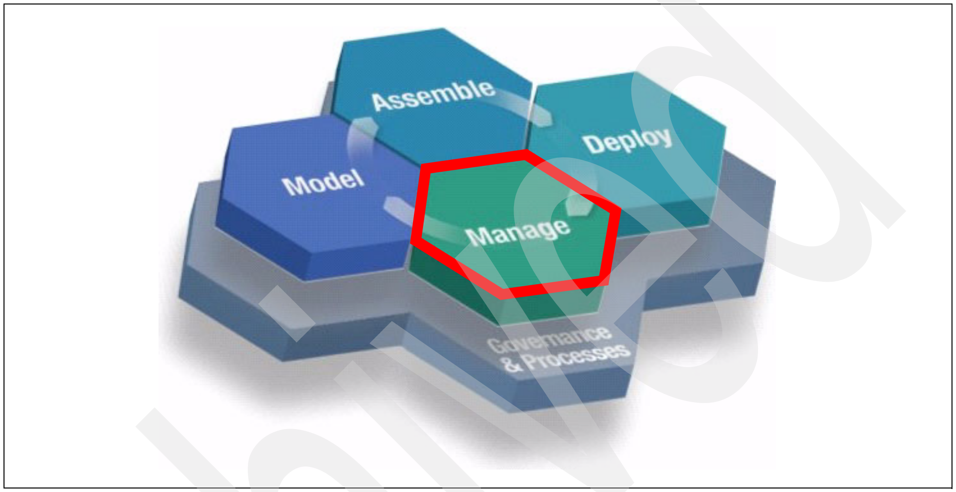


Figure 9-1 IBM SOA Foundation life cycle

The final stage of an iteration in a life cycle is the management stage. A service needs to operate with the correct quality of service (QoS). The following four steps relate to operating with the correct QoS:

- Managing services

  Management of services includes service consumers and service providers. Service consumers invoke services, while service providers expose application functions to be consumed. When speaking in general terms, management of services refers to a range of service technologies.

  Key elements of managing the services layer are:

  – Understanding how services relate to each other and to the IT infrastructure, as well as how the service relates to the business process layer

  – Controlling the message flow in the service environment through management mediations, such as log, filter, and route. The message flow often spans the architectural layers.

  – Centralizing the services management policy

– Defining business-related IT goals.

- Managing transaction performance

  Managing and monitoring the end-to-end transaction performance is a key measurement for service level agreements (SLAs). Managing the transaction performance is also a very useful analysis and troubleshooting tool. For example, you can proactively detect a slowdown in performance before it becomes a critical problem that stops the transaction from completing. Also, a view of the transactional performance can be very helpful for troubleshooting to isolate the resources that are not performing or are failing:

  Managing transaction performance includes:

  – Understanding the performance of a service and the decomposition of transactions with specific metrics for individual requests

  – Providing the relationship between service requests and the implementation artifacts, such as J2EE beans and JDBC™ requests

- Managing the supporting middleware

  Many of the resources used by services are found in the middleware. For example, WebSphere Application Server application servers are used to host J2EE and Web services applications. DB2® Universal Database™ is used to host databases. IBM WebSphere MQ is used for messaging. Each of these resources should be managed and monitored.

  Managing the supporting middleware includes the following concepts:

  – Understanding the health of the infrastructure that supports the services

  – Correlating problems in the services to infrastructure issues, such as a queue filling up or an exhausted thread pool

- Managing the operational systems

  SOA environments are built using real resources, and those resources must also be managed. Managing the operational systems includes understanding the health of the infrastructure that supports the services.

The SOA management solution has to be able to give feedback for starting the next life cycle of a service.

9.2 IBM service management

All needed tools, data, and processes for SOA management are provided by *IBM Service Management*. The ongoing management, enhancement, and redeployment of services involve many people, processes, and pieces of information. An organization needs ways to connect them all. Continuing the momentum of IT Service Management (ITSM), the strategy has been renamed IBM Service Management, or ISM for short. The strategy has four key elements:

- IT process manager products (PMPs)
- Service management platform (CCMDB)
- Operational management products (OMPs)
- IBM best practices in the form of services

IBM Service Management (Figure 9-2) offers a common data model that facilitates real-time information sharing. Embedded in the usage of best practices (for example ITIL-aligned workflows), IBM Service Management is based on information delivered by IT Operational Management Products for:

- Business application management
- Service, network, and device management
- Storage management
- Security management

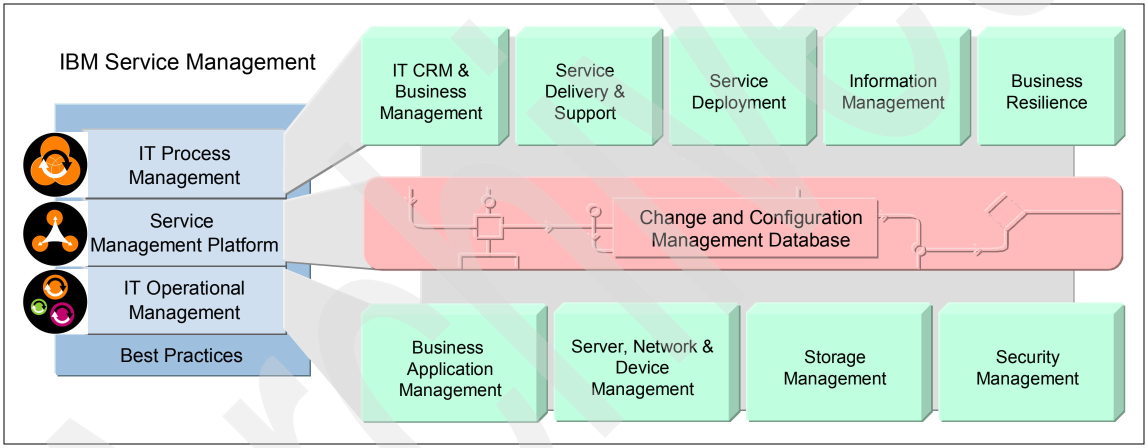


Figure 9-2 IBM service management overview

The open and standards-based Service Management Platform, the Change and Configuration Management Database (CCMDB), is described in more detail in 9.4.2, "IBM Tivoli CCMDB" on page 269.

The available information is used by management processes belonging to:

- IT CRM and Business Management
- Service Delivery and Support
- Service Deployment
- Information Management
- Business Resilience

Based on the standard processes described in ITIL (see "ITIL" on page 47), the following processes are available for implementation:

- Change Process Manager (in CCMDB)
- Configuration Process Manager (in CCMDB)
- IBM Tivoli Release Process Manager
- IBM Tivoli Availability Process Manager
- IBM Tivoli Storage Process Manager
- IBM Tivoli Capacity Process Manager

Additional management processes are in development or planned.

The integration of change, configuration, and release management and the integration of the service desk with the embedded processes for incidents and problem management bring additional information into IBM Service Management. An overview of the *IBM Service Management Platform* is shown in Figure 9-3.

IBM Service Management solutions are available at:

http://www.ibm.com/software/tivoli/governance/servicemanagement/

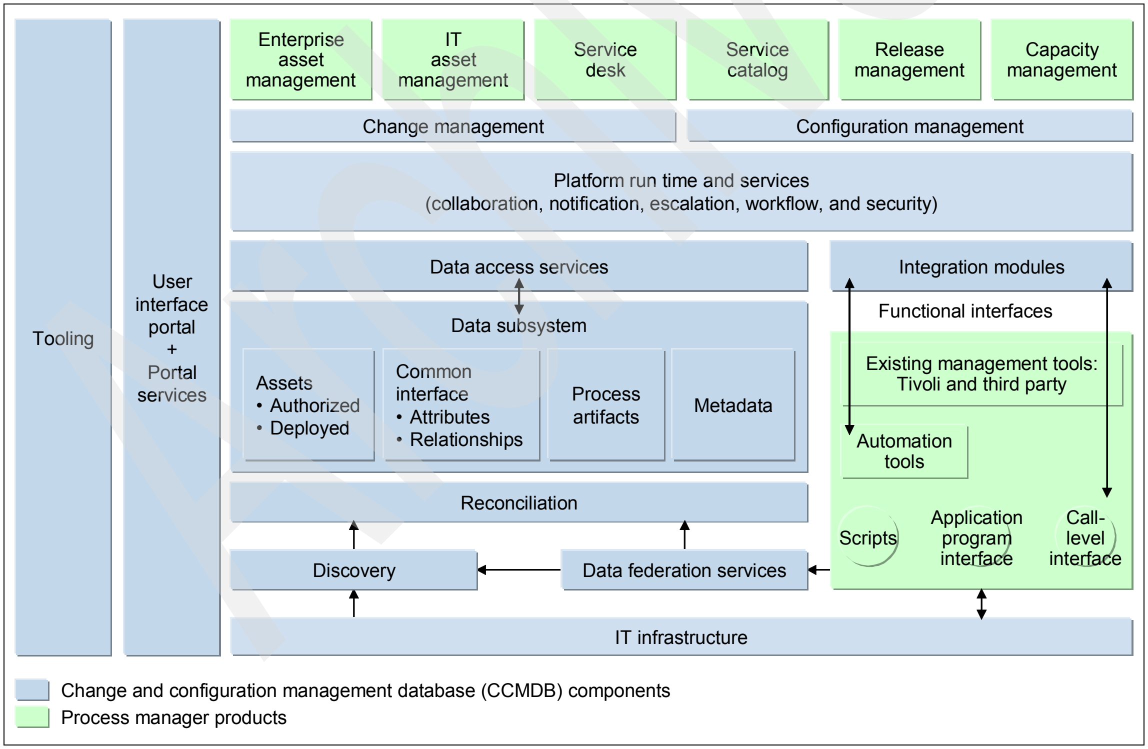


Figure 9-3 IBM Service Management Platform

9.3 SOA management solution

Based on the adoption level of the SOA life cycle, we discuss the following interaction tasks here:

- Observe service
- Capture diagnostics
- Assess service performance
- Understand service relationships
- Understand system structure

These interactions are covered by the two sides of SOA management solutions:

- The management of IT systems that implement the business solution
- The management of the IT system's effect on a business

Both of these domains are built out of single services, which combine IT systems with business processes or composite applications. Each business service is able to run on one or more IT systems based on its quality of service (QoS). Figure 9-4 gives you an overview of the connections between the business and the IT domain of an SOA solution.

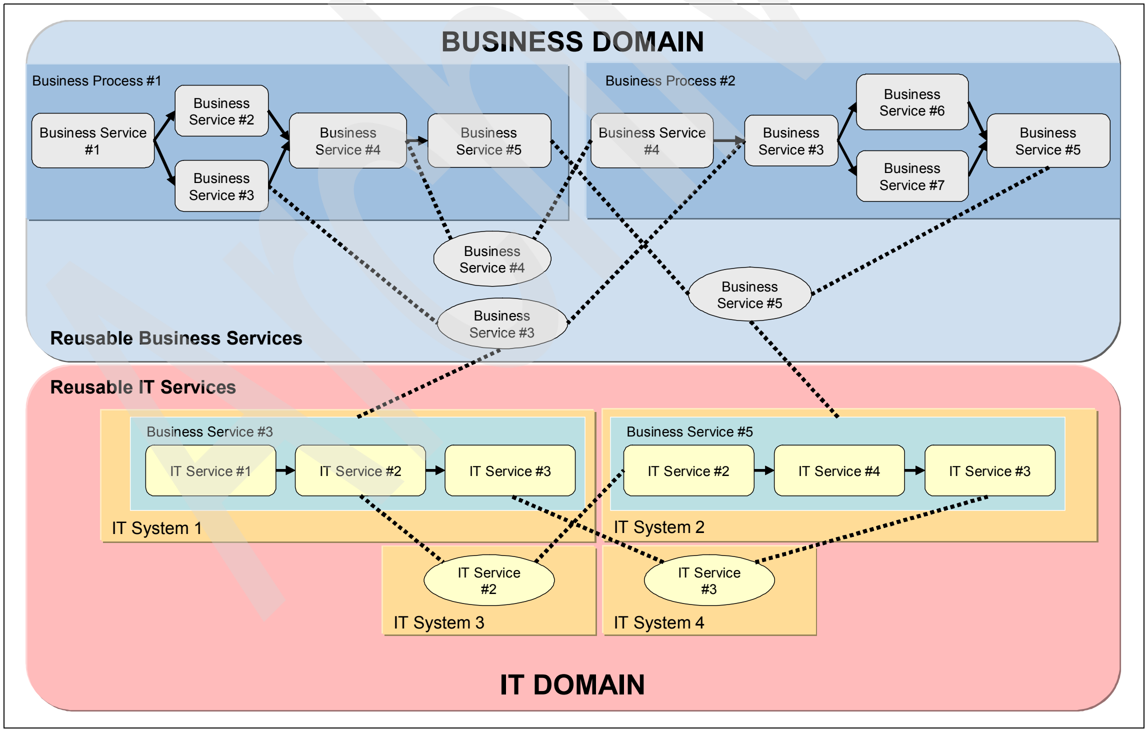


Figure 9-4 Services in the business and IT domain

The thoughts related to an SOA management solution are the thoughts for managing IT systems with the focus of ensuring the productive and efficient use of these computing resources, for example:

- Ensuring systems are up and running
- Identifying problems and resolving them
- Making sure users get the performance and availability they need
- Balancing the use of resources to meet service level agreements

In addition to the traditional role of systems management, there is a correlation between the business design and IT with the resources that map to that design. SOA bridges the business design and the IT implementation and based on this, it is possible to move from the modeling space through the assemble stage into the development and deployed environment.

With service management, it is possible to detect whether:

- A problem within the IT system is affecting the business
- Business goals and objectives are being met as efficiently as possible
- The source of missing goals is based on constraints in the IT system or in the business design

9.3.1 Layers of SOA management

SOA management can be thought of in the following three layers:

- *Business Service Management* provides for service level planning, business impact monitoring, and prioritization of event management.

  Business service management (BSM) shows the connection between the process management and the IT service management.

  The target of BSM is to get a better relationship between business and IT. The foundation for BSM is a real-time IT services monitoring, reporting, and management solution. The data of this solution has to map with the business processes to build a data repository, which is the configuration management database (CMDB).

  The IBM solution, IBM Tivoli Business Service Manager, allows definition of business service objects in a tree-like manner. Each service object has a color-coded state associated with it. These states are calculated either from the state of its descendents or from the effect of an external event. The service object can represent an actual IT resource or an abstract entity. A typical abstract entity can depict a geographical region, business function, application system, or a collection of items.

IBM Tivoli Business Service Manager can be used as:

- An operational tool for alerting operators of impending problems regarding a specific function or business process
- A prioritization tool to understand the business impact of the outage of an IT resource
- An executive tool that shows the state of a business function
- A service level analysis tool to calculate the net effect of different outages and failures to the overall service level objective

For more information about business service management, refer to:

- *IBM Tivoli Business Service Manager V4.1*, REDP-4288
- *IBM Tivoli Business Service Management solutions* are available at:

  http://www.ibm.com/software/tivoli/solutions/bsm/

- *Composite Application Management* provides support for securing the SOA environment, flow content analysis, user response time monitoring for service requests, service problem diagnosis, and application trace information that you can then pass back to your development environment.

  The IBM Tivoli Composite Application Manager (ITCAM) product family consists of the following products:

  - ITCAM for SOA: Monitor and manage the SOA services layer.
  - ITCAM for Response Time Tracking: Proactively recognize, isolate, and resolve transaction performance problems.
  - ITCAM for WebSphere: Isolate the root cause of bottlenecks in a WebSphere application runtime environment.
  - ITCAM for CICS Transactions: Capture data from CICS systems to be analyzed in ITCAM for Response Time Tracking and WebSphere.
  - ITCAM for IMS™ Transactions: Capture data for IMS systems to be analyzed in ITCAM for Response Time Tracking.

  For more information about composite application management, refer to:

  - *IBM Tivoli Composite Application Manager V6.1 Family Installation, Configuration, and Basic Usage*, SG24-7151
  - *Large-Scale Implementation of IBM Tivoli Composite Application Manager*, REDP-4162
  - *Solution Deployment Guide for IBM Tivoli Composite Application Manager for WebSphere*, SG24-7293
  - *Best Practices for SOA Management*, REDP-4233.

- *IBM Tivoli Composite Application Management solutions* are available at:

  http://www.ibm.com/software/tivoli/solutions/application-management/

- *Resource Management* enables orchestration, provisioning, infrastructure health monitoring, and event automation.

  The main component or function of resource management in an SOA environment is provisioning. *Provisioning* is the end-to-end capability to automatically deploy and dynamically optimize resources in response to business objectives in heterogeneous environments. Provisioning helps to respond to changing business conditions by enabling the ability to dynamically allocate resources to the component applications, processes, or services that most need them, as driven by business policies.

  Provisioning of individual IT services, such as identities, storage, servers, applications, operating systems, and middleware is a critical step to orchestrate the entire environment, enabling it to respond to business needs on demand. Provisioning focuses on the self-configuring, dynamic allocation of individual elements of the IT infrastructure so that identities, storage, or servers are provisioned as business needs dictate.

  For more information about composite application management, refer to:

  - *Deployment Guide Series: IBM Tivoli Provisioning Manager Version 5.1*, SG24-7261
  - *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117
  - *A Practical Guide to the IBM Autonomic Computing Toolkit*, SG24-6635
  - *IBM IT Service Management Orchestration and Provisioning solutions* are available at:

    http://www.ibm.com/software/tivoli/solutions/orchestration-provisioning-management/

These three layers of management are integrated by a common understanding of the model for service-oriented computing and a central information model that describes the configuration of the IT systems, the SOA services, and the applications that have been deployed to that environment.

A big part of SOA management includes provisioning the deployment environment. Key to this is establishing a user registry, administrative policy and procedures, and a configuration scheme that will support the physical architecture defined by the Enterprise Architect - matching the physical architecture to the policies and procedures for setting and expanding the capacity of the system as demand increases for SOA services.

Given the increasing complexity and heterogeneity of IT systems in general and given the importance of IT systems to operate the business, it is essential that IT systems are as resilient as possible. This requirement generates the need for autonomic systems: IT systems with the capability to manage themselves and to be self-configuring, self-healing, self-tuning, and self-protecting. Autonomic computing becomes even more important in an SOA environment.

Services are inherently loosely-coupled, logically distributed, and highly shared and reused. A failure in one service can impact many other services. However, unlike monolithic systems, there might be a delay before these impacts are known. You might not detect a failure until you go to use the service. You do not want to find out about a failure for the first time when your business users call you to say they cannot get their job done.

An SOA infrastructure needs to be able to identify its own problems and automatically initiate processes to fix them - even without having to wait for an administrator to figure that out. These systems need to monitor themselves and automatically adjust resource consumption, queue sizes, and priorities, degrees of parallelism, and so on, to ensure the resources of the system are used efficiently and to route around bottlenecks.

The system needs to automatically reconfigure itself as needed to meet the service level objectives. This is important not just because it will reduce your administrative overhead, but because human beings typically cannot respond quickly enough to handle the type of dynamics that occur in business marketplaces on a daily, hourly, or even minute-by-minute or second-by-second basis.

The typical monitor, analyze, plan, and execute loop of autonomic computing requires integral support in the SOA runtime infrastructure, tools, and management services of the SOA Foundation.

9.4 Configuration Management Database

The Configuration Management Database (CMDB) is defined by the IT Infrastructure Library (ITIL) as the database where Configuration Items (CIs) are stored. *CIs* are data about managed resources, such as computer systems and application software, process artifacts, such as incident and change records, and the relationships among them.

The CMDB represents the integration of several enterprise applications and their data. Given the complexity and scale of the integration involved, it is critical that organizations take a phased approach to build the CMDB. Without a careful process, identifying an optimal starting point and then following a phased integration approach to integrate other applications, creating the CMDB can be very expensive and time-consuming.

At the core of ITIL are the IT applications and their supporting infrastructure components. Therefore, the ideal starting point to create the CMDB is a database of the IT application and its IT infrastructure. Today, there are several application discovery solutions that provide an unparalleled level of visibility into how the infrastructure actually delivers the applications on which the business relies. These solutions automatically create and maintain cross-tier maps of the application and its supporting infrastructure. These application maps also include the in-depth configuration values of the supporting infrastructure components and their runtime dependencies. After it is created, this database can be integrated to the other enterprise applications, including:

- Process applications
- Organizational data applications

Besides discovery, the CMDB must provide the following critical functions, which distinguish the CMDB from other tools:

- *Reconciliation* ensures the data is coalesced, avoiding duplicates and enabling the matching of configuration items from different sources.
- *Federation* brings in multiple data sources directly and also by linking to sources.
- *Mapping and visualization* enable a peer-to-peer and hierarchical view of the CIs.
- *Synchronization* ensures the same version of the truth across integrated systems.
- *Access controls* ensure only the right administration changes are made to the schema and that access is monitored at the CI level.

Tools marketed as CMDBs that do not have all four critical functions will require significant human resources to provide the additional functions.

Reconciliation

Reconciliation is the ability to rationalize the same instance of a CI or component that might come into the CMDB from multiple sources. One discovery tool might see a UNIX® device by a host name, another might discover it by an IP address, and additional technology by its Media Access Control (MAC) address. Reconciliation ensures that there is only one instance of this server with the correct configuration data represented in the CMDB.

The most challenging aspect of reconciliation is to be able to determine that the same instance of a component or CI coming from different sources might have different naming conventions. Reconciliation also must include a capability that checks the relationship for integrity to ensure that the managed linkages are both semantically and actually accurate. Therefore, the CMDB must ensure that the reconciliation engine can determine that multiple identities are actually the same component or CI to enable the CMDB to be the hub between different management tools.

Federation

Federation enables multiple data sources to feed some level of data or just link data stores and configuration repositories into the CMDB while the individual IT domain sources continue to maintain detailed configuration information about an infrastructure component. It is impossible to store and manage all configuration data of an enterprise in a single CMDB.

Federation is not just integration (as performed in tools for discovery data) or the ability to bring data into a new data store (such as integrating device information into an asset tool). It is the ability to bring multiple data sources into a coalesced view where the various feeds come together to represent a view with relationships across components.

Mapping and visualization

Mapping and visualization provide the ability to illustrate logically and physically the peer-to-peer and hierarchical relationships between CIs. IT service management tools must discover applications, underlying servers, storage devices, and load balancers, as well as switching fabric to show peer-to-peer and hierarchical relationships across them. In addition, the visualization of the direction of the relationships is core to facilitate IT in managing business services.

Tools (such as Configuration Discovery and Tracking) can run as stand-alone dependency mapping discovery solutions. They build the foundation for a CMDB to be able to visualize relationships across all CIs from a variety of federated data sources.

Mapping and visualization also include reporting and analysis capabilities, such as running reports to see which applications reside on specific resources or components.

Synchronization

Synchronization is the ability to update the CMDB with approved changes, as well as to identify changes that are not approved. After a baseline is established, comparisons are made against various sources. If inappropriate changes are detected, a notification is triggered to a change management workflow to alert the appropriate IT domain to investigate and potentially remedy the drifted configuration. Synchronization results in achieving the goals of closed-loop change control.

A CMDB enables the ability to provide input into the risk and impact analysis of a planned change with a view of the interdependencies of one IT component with respect to another (such as helping to understand the impact of a vulnerability in one application on all of its associated components). This capability will also be able to help IT organizations quickly identify when a system is not in its desired state, although the CMDB is working in conjunction with its federated data sources to get the system into its desired state.

Access controls

Access controls ensure that only the appropriate roles (human beings, as well as tools and systems) have read and write access to the information. This is a bigger requirement than you might think at first glance. Because of the federated nature of the information, updates to the CMDB might not update only the central store, but also likely cascade to the trusted source, or perform the reverse and report on the discrepancies.

9.4.1 CMDB federation

It is often challenging to integrate the Management Data Repositories (MDRs) of one or more vendors, because there is no defined standard. The CMDB Federation specification (CMDBF) is a joint deliverable from IBM and five partners: BMC, CA, Fujitsu, Hewlett-Packard, and Microsoft, to address this issue.

The CMDBF defines a minimal data model. Related specifications for data formatting and modeling are:

- Service Modeling Language (SML), an extension to the XML Schema 1.1
- SML Interchange Format (SML-IF), an interchange format for SML
- Common Model Library (CML), which will be a set of specifications

The definition of a CMDB in the context of the CMDBF is based on the definition described in the IT Infrastructure Library (ITIL):

> *A database that tracks and records configuration items associated with the IT infrastructure and the relationships between them.*

The CMDBF extends this base definition to federate any management information that an administrator configures, as long as the information and source data repositories comply with the specification’s patterns, schema, and interfaces (Figure 9-5 on page 268).

For example, the federated CMDB might include the observed configuration, as well as the authorized configuration; the configuration history; proposed or projected future states; process artifacts, such as requests for change, incident records, and audit records; status change events; relationships among this data.

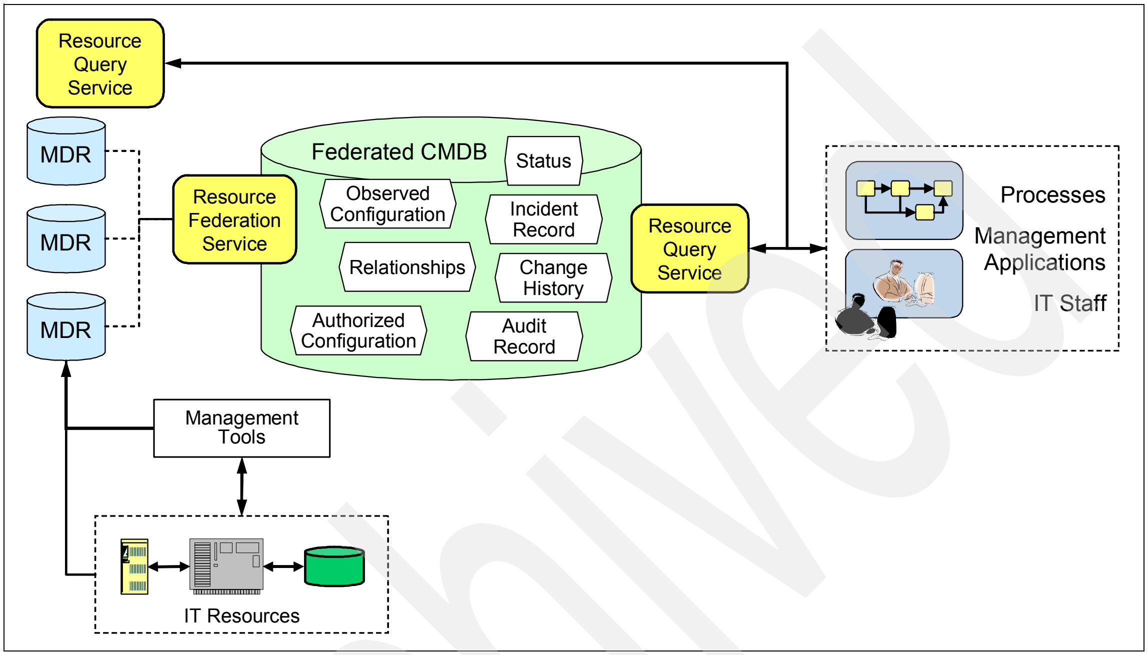


Figure 9-5 CMDBF architectural overview

One CMDB and a number of management data repositories (MDRs) comprise the federated CMDB. Each MDR can be viewed as a source for some set of data about IT resources, process artifacts, and the relationships between them. Examples of MDRs include an asset management repository, a network management database, and a service desk managing incident and problem records. A typical implementation of the federated CMDB will normalize data, arbitrate between multiple sources that provide overlapping data, and reconcile resource names to recognize situations in which different names refer to the same resource.

The CMDB Federation Workgroup is available at:

http://www.cmdbf.org/

9.4.2 IBM Tivoli CCMDB

IBM Tivoli Change and Configuration Management Database (CCMDB) is the IBM product that covers this functionality and that is a key component of the IBM IT service management strategy.

With IBM Tivoli Change and Configuration Management Database Configuration Discovery and Tracking V1.1 and its predecessor product, IBM Tivoli Application Discovery and Dependency Manager (TADDM), IBM allows a jump-start ITIL initiative. By first using agentless auto-discovery technologies, it is possible to quickly and continuously populate a federated CMDB. This CMDB includes an inventory of physical resources (network devices, hosts, and storage devices), applications (software services), and the interdependencies of these resources and applications.

The IBM Tivoli CCMDB delivers a federated view of all your enterprise's IT data, including information about hardware, software, and the relationships between them. In addition, it provides a process foundation for the delivery of value-added solutions for release management, availability management, and storage management.

The IBM Tivoli CCMDB also integrates IT service functions into a unified, automated infrastructure management platform, which helps to:

- Consolidate information between disparate IT environments
- Create synergy between different IT service management functions
- Optimize the management of IT service demands
- Maximize IT performance and return on investment (ROI)

An overview of IBM Tivoli CCMDB is shown in Figure 9-6 on page 270.

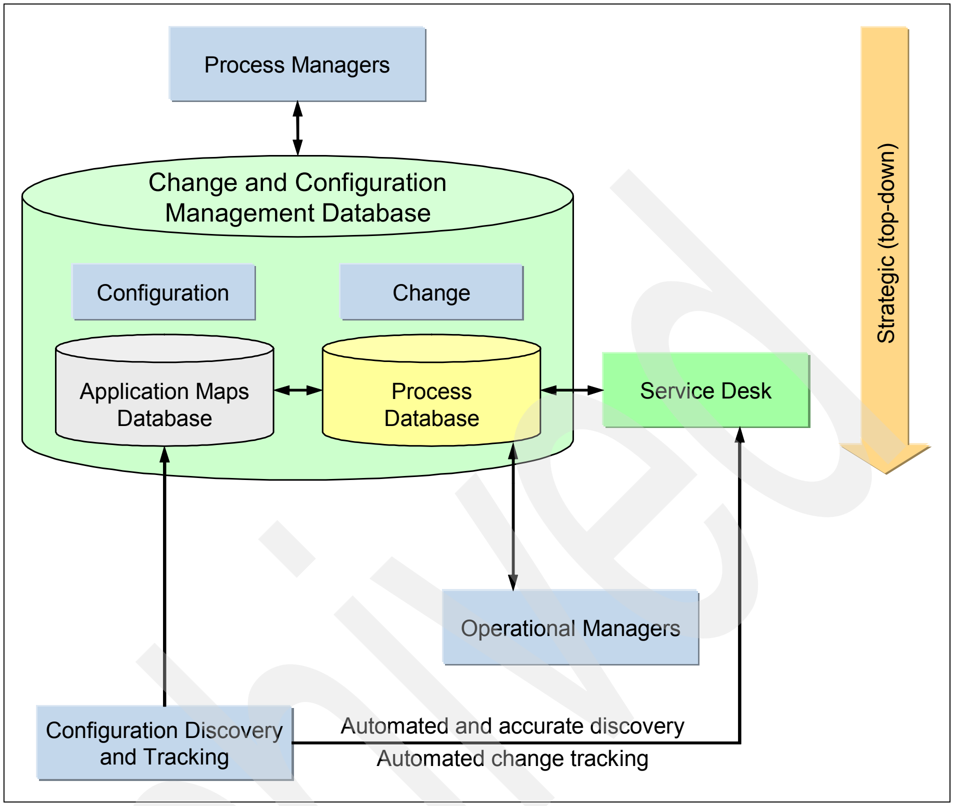


Figure 9-6 Automate processes through the integration of operational managers

A CMDB strategy is central to the IT Service Management framework implementation described in the IT Infrastructure Library (ITIL) documents. ITIL outlines two main areas of this framework:

- Service support to enable effective delivery of IT services
- Service delivery that outlines the management of these services

Projects require this Configuration Item (CI) data in order to begin to take advantage of these best practice frameworks, therefore, the need for the CMDB.

IBM Tivoli Process Managers and Operational Managers

By adopting a federated CMDB approach, IT organizations can use repositories throughout their environment through integration and the utilization of open interfaces to share data. However, this is just a portion of the information needed and does not begin to give the organization the continuous application infrastructure data required to build their overall best-practice strategies.

To further assist IT managers in their deployment of ITIL best practices, IBM has developed IBM Tivoli Process Managers. A collection of predefined, automation packages, IBM Tivoli Process Managers outline the people, resources, and information needed to create IT processes that are repeatable, measurable, and efficient.

For more information about IBM Tivoli CCMDB, refer to the *Deployment Guide Series: IBM Tivoli Change and Configuration Management Database Configuration Discovery and Tracking v1.1*, SG24-7264.

Part 3

Appendixes

A

Product offerings and descriptions

This chapter discusses the products that are available in assisting an organization with the service-oriented architecture (SOA) life cycle, governance, and management. It includes:

- Overview of IBM software portfolio
- Products for service development
- Products for service deployment
- Products for service management

Overview of IBM software portfolio

IBM offers five pillars of software brands (Figure A-1).

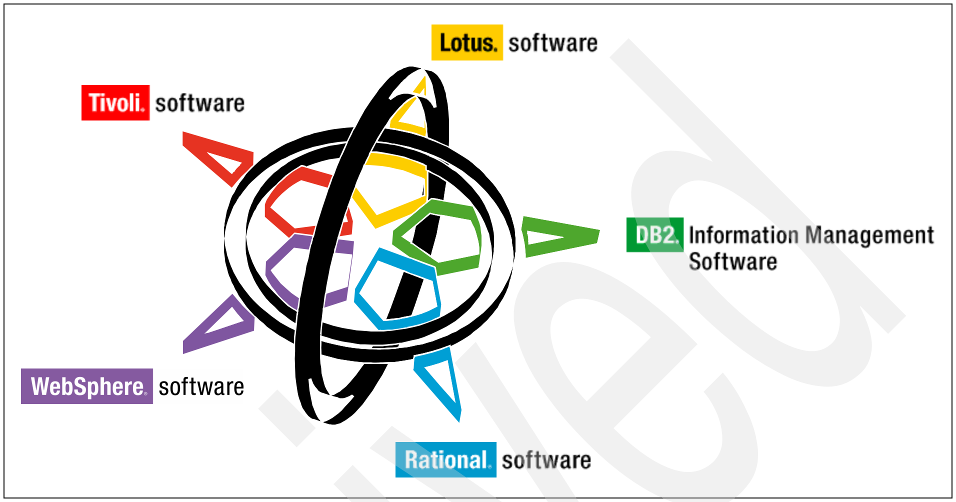


Figure A-1 IBM software brand compass

Tivoli software

The Tivoli software builds the base for an effective combination of business and IT services in an SOA environment. The Tivoli Service Management enables clients to better manage their infrastructure, operations, and IT processes and to more effectively deliver services aligned to business goals. There are solutions for the following areas available:

- Asset management

  Achieve greater efficiency in asset management by managing all your asset types on a single platform.

- Business application management

  Manage composite applications and optimize application performance and service levels.

- Security management

  Ensure compliance to identity and access control policies for IT resources and services.

- Server, network, and device management

  Optimize performance and automate the provisioning of IT infrastructure resources.

- Service management

  Innovation, execution, and leadership for enterprises to optimize and manage the business of IT.

- Service provider solutions

  Ensure critical services are performing to the highest standards.

- Storage management

  Backup, restore, protect, and optimize your storage infrastructure and data.

You can obtain additional information from:

http://www.ibm.com/software/tivoli/

Lotus software

Lotus® software focuses on the presentation layer in an SOA reference architecture and enables businesses to communicate, collaborate, and increase productivity at a highly secure level. There are solutions for the following areas available:

- Application design and development

  Increase user productivity and business flexibility with custom solutions for Lotus software.

- Dashboards and business solutions

  Deploy role-based work environments and improve time-to-value with dashboards, scorecards, and composite applications.

- E-mail, calendaring, and collaborative applications

  Provide more than just messaging with Lotus Notes® and Domino® built-in collaboration tools and application platform.

- Instant messaging and Web conferencing

  Collaborate real-time with integrated Unified Communications and Collaboration (UC^2) software and solutions.

- Mobile and wireless

  Extend enterprise business processes, applications, and information to mobile and wireless devices and clients.

- Social software

  Gather and exchange information through professional networks and build communities of experts to help execute tasks faster.

- Team collaboration, content management, and e-forms

  Create, organize, share, and manage business content to provide the right information to the right people fast.

You can obtain more information from:

http://www.ibm.com/software/lotus/

DB2 Information Management software

In an SOA environment, Information Management software integrates data and enterprise content to leverage information on demand. There are solutions for the following areas available:

- Database servers

  Build the foundation for information on demand.

- Database tools

  Enhance IMS and Information Management software performance.

- Dynamic data warehousing and business intelligence

  Gather, manage, and analyze data to generate insight.

- Enterprise content management

  Manage volume and capture the value of unstructured content.

- Information platform and solutions

  Leverage information for innovation combining market-leading Information Integration and Master Data Management technologies.

You can obtain more information from:

http://www.ibm.com/software/data/

WebSphere software

The IBM WebSphere software palette builds the platform for governing software and systems delivery. There are solutions for the following areas available:

- Application and transaction infrastructure

  Build, deploy, and run applications in a proven, secure, and flexible environment.

- Application transformation

  Leverage existing applications and information in new business processes.
- Business process management

  Improve business responsiveness with end-to-end integration across and beyond the enterprise.
- Commerce

  Automate and integrate online marketing and sales processes across multiple channels.
- Mobile and speech middleware

  Extend access to business processes, applications, and information to anyone anywhere.
- Portals

  Deliver a point of personalized interaction with applications, content, processes, and people.
- Product information management

  Provide a single, integrated source of product information for use across the value chain.

You can obtain more information from:

http://www.ibm.com/software/websphere/

Rational software

The platform for governing software and systems delivery is built by Rational software. There are solutions available for the following areas:

- Architecture management

  Model, design, and rapidly build resilient architectures for SOA, systems, and applications.
- Change and release management

  Improve software delivery and life cycle traceability from requirements through deployment.
- Process and portfolio management

  Align business goals, best practices, and projects for improved productivity and predictability.

- Quality management

  Ensure software functionality, reliability, and performance in development and production.

You can obtain more information from:

http://www.ibm.com/software/rational/

Products for service development

This section describes products used during the service development life cycle and highlights their key benefits for SOA projects.

Rational Software Development Platform

The Rational Software Development Platform is an integrated set of products that are based around the Eclipse V3.2 framework for the support during the development life cycle. It provides teaming capabilities optimized for a number of roles in a project team, including business analysts, architects, developers, testers, and deployment managers.

Figure A-2 shows several of the products in the Rational Software Development Platform for the model and assemble SOA life cycle and the associated project roles that might use the products.

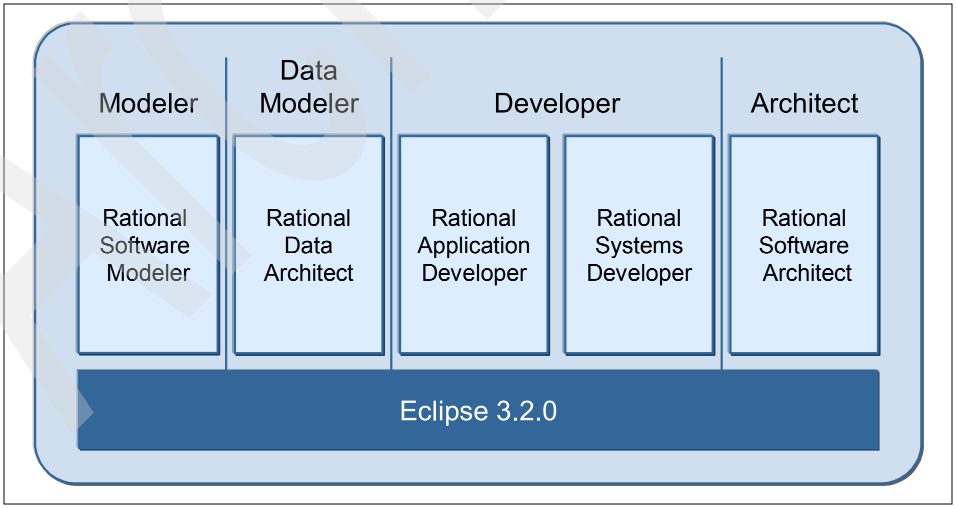


Figure A-2 Rational Software Development Platform model and assemble products

A brief description of each of the Rational Software Development Platform products is included to identify key features.

Rational Software Modeler

Rational Software Modeler V7.0 is a Unified Modelling Language (UML) 2.0 modeling tool for use in system analysis and design. It includes the ability to apply patterns and transformations leveraging common design patterns and best practices for sound system architectures. The separation of business and application logic is supported in Rational Software Modeler using Object Management Group's (OMG) Model Driven Architecture (MDA). This allows multiple levels of models to be defined with user-defined transformations between models and code. Models generated in Rational Software Modeler can also be used with Rational Software Architect and Rational Systems Designer products.

You can obtain additional information about Rational Software Modeler from:

http://www.ibm.com/software/awdtools/modeler/swmodeler/index.html

Rational Data Architect

Rational Data Architect V7.0 is used by data architects or modelers in the design of relational and federated databases to understand information assets and their relationships and for streamlining database projects. It provides features to discover, model, visualize, and relate heterogeneous data assets. Visual models can be represented using Information Engineering (IE) or UML notation to create logical and physical data models. Rational Data Architect can help you with governing and assisting in corporate naming standards by enabling specification of valid names and abbreviations in glossaries to find nonconformance in the models or actual databases.

You can obtain additional information about Rational Data Architect from:

http://www.ibm.com/software/data/integration/rda/

Rational Systems Developer

Rational Systems Developer V7.0 is an integrated design and development tool for systems architects to model complex systems. It incorporates the design and UML 2.0 modeling functionality of Rational Software Modeler and an Integrated Development Environment (IDE) to create software level system components in Java Standard Edition (SE), C/C++, and CORBA. UML-based transformations are included for the generation of Java, C++, and CORBA, including the generation of Logical Data Models (LDMs) that is supported in Rational Data Architect. Tooling is provided for the development of custom transformations to target any type of implementation output from UML. Rational Systems Developer also includes features for analysis, such as a structural review of Java code allowing visualization of existing design patterns and automatic detection of structural anti-patterns.

You can obtain additional information about Rational Systems Developer from:

http://www.ibm.com/software/awdtools/developer/systemsdeveloper/index.html

Rational Application Developer

Rational Application Developer V7.0 is an integrated development tool for Java Standard Edition (SE), Java Enterprise Edition (J2EE), portal, Web, Web services, and SOA applications. It provides J2EE V1.2, V1.3, and V1.4 compatibility with full support for Enterprise JavaBeans™ (EJB™), message driven beans, EJB Query Language, Container-Managed Persistence (CMP), Web archive (WAR), and enterprise archive (EAR) deployments. Full support for Web services is provided using existing Web Services Description Language (WSDL) and Web Services Inspection Language (WSIL) files for automatic generation of code or development of new Web services using the visual editors. Rational Application Developer provides support for virtually all aspects of Web development using HTML, CSS, JavaServer™ Pages™ (JSP™), JavaServer Faces (JSF), and servlets.

Full integration is supported for WebSphere Application Server V5.1, V6.0, and V6.1, including WebSphere Portal Server V5.1 and V6.0 for the development, testing, and debugging of applications. For quality control of code, Rational Application Developer includes an automated code review feature supported by more than two hundred J2EE coding standards and best practices to application code highlighting violations and suggested resolutions.

You can obtain more information about Rational Application Developer from:

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

Rational Software Architect

Rational Software Architect V7.0 provides an integrated design and construction environment to assist software architects to understand, design, manage, and evolve enterprise solutions across a team. Support for modeling using Unified Modelling Language (UML) and development using Java Standard Edition (SE) and Java Enterprise Edition (J2EE) is included in the application. It combines the functionality of Rational Software Modeler, Rational Application Developer, and key analysis and design functionality from Rational Systems Developer in a combined solution. Additionally, Rational Software Architect includes the support of:

- J2EE rule analysis and reporting

- UML class diagram enhancements for Enterprise Java Beans (EJBs), data tables, and sequence diagrams of method bodies
- UML code to code transformations for EJB, Java, C++, CORBA IDL, and XSD
- Process guidance for the developer and the architect to assist them in developing Java and C++ applications using Rational Unified Process (RUP)

You can obtain more information about Rational Software Architect from:

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

Key benefits for SOA projects

The Rational Software Development Platform provides a number of key benefits:

- Common standardized modeling, development, testing, and deployment environment
- Eclipse pluggable framework, which is the feature to include corporate-based standards and interfaces

WebSphere Business Modeler

WebSphere Business Modeler V6.0.2 is a modelling tool for use in the visualization, comprehension, and documentation of an organization's business processes. It provides the ability to model, assemble, and deploy business processes and monitor Key Performance Indicators (KPIs) specified in the model to optimize the processes. WebSphere Business Modeler is based on the Eclipse V3.0.2 platform.

The WebSphere Business Modeler family comes in various product options:

- WebSphere Business Modeler Basic

  Provides basic modelling capabilities to model, validate, and document business processes and includes:

  – A reporting tool for presenting information in the model
  – Copy and paste of process diagrams to Microsoft Powerpoint and Microsoft Word
  – Publish swimlane models to WebSphere Business Modeler Publishing Server
  – Import business services, such as Web Services Description Language (WSDL) and business objects, such as XML Schema Definition (XSD)

- WebSphere Business Modeler Advanced

  Includes the features of WebSphere Business Modeler Basic with additional capabilities, such as:

  – Simulation and analysis of business processes
  – Integration with WebSphere Process Server using Business Process Execution Language (WS-BPEL)
  – Integration with WebSphere MQ Workflow using the Flow Definition Language (FDL)
  – Ability to define business level measures that are associated with the process that allow it to be viewed using WebSphere Business Monitor in the runtime environment

- WebSphere Business Modeler Publishing Server

  Used for publishing business processes created in WebSphere Business Modeler. Authorized users can view and provide comments on the business processes via a Web browser. These comments and responses can be tracked by the publisher of the business process and exported to allow the business process to be updated.

Key benefits for SOA projects

WebSphere Business Modeler provides a number of key benefits for SOA projects:

- Develops a regulatory compliance through well documented auditable processes
- Defines business level Key Performance Indicators (KPIs) for monitoring a company's processes

You can obtain more information about WebSphere Business Modeler from:

http://www.ibm.com/software/integration/wbimodeler/index.html

WebSphere Integration Developer

WebSphere Integration Developer V6.0.2 is a development tool used for the assembly of SOA-based services that choreograph the use of business processes. The target deployment platform includes WebSphere Process Server V6.0.2 and WebSphere Enterprise Service Bus V6.0.2 and focuses on providing a tool for the role of the integration developer. This platform complements WebSphere Business Modeler V6.0.2 and can be used with Rational Application Developer V6 and Rational Software Architect V6. WebSphere Integration Developer is based on the Eclipse V3.0.2 platform.

The service components supported in WebSphere Integration Developer for WebSphere Process Server include:

- Business processes and human task support using Business Process Execution Language (BPEL)
- Business state machines for modeling heavily event-driven business process scenarios
- Business rules that provide support for rule sets (if-then rules) and decision tables
- Supporting components:
  - Interface maps for mapping between interfaces
  - Business object maps to translate from one business object to another
  - Relationships for access of data sets in back-end systems
  - Selectors for dynamic invocation based on various rules

Key benefits for SOA projects

WebSphere Integration Developer provides a number of key benefits for SOA Projects:

- Encourages reuse of existing assets as service components
- Provides tooling for use of standard interfaces between SOA components
- Focuses on the assembly phase of the SOA life cycle

You can obtain more information about WebSphere Integration Developer from:

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

Progress DataXtend Semantic Integrator

Progress DataXtend Semantic Integrator is a solution for solving the challenges of the business integrity of data. Graphical tools are provided to operate with development environments utilizing XML in a J2EE application server environment. The product includes a set of visual tools to support the development process in Progress DataXtend Semantic Integrator Designer of defining the data mapping and a runtime component that is used in deployment in Progress DataXtend Semantic Integrator Engine.

The product focuses on the use of a common model for data for use at the enterprise level and the ability to map between a data source and a data target. A number of industry or enterprise models and other sources are supported through an import process provided in Progress DataXtend Semantic Integrator, including but not limited to:

- TeleManagement Forum:
  - Telecommunications Shared Information Data (DID) Model
- Open Applications Group Integration Specification (OAGIS)
- RosettaNet
- Web Services Definition Language (WSDL) files
- Database schemas

The Progress DataXtend Semantic Integrator Designer product is used during the development process to define the mappings between the common model that is used and the data source and data service models. Validation rules for enforcing constraints are derived from imported XML schemas with the ability to define custom attributes and validation, as well as business rules without the addition of code.

The runtime component, Progress DataXtend Semantic Integrator Engine, output is Java running as a stateless service and is used to dynamically convert and validate data prior to the submission to back-end systems.

Key benefit for SOA projects

Progress DataXtend Semantic Integrator provides the use of a common data model that leverages industry or common standards.

You can obtain more information about Progress DataXtend Semantic Integrator from:

http://www.progress.com/dataxtend/dataxtend_si/index.ssp

Rational ClearQuest

Rational ClearQuest V7.0 is a solution for use during the application development life cycle and for the management of change, test planning, and execution. The functions that Rational ClearQuest provide include:

- Defect creation and management
- Enhancement requests
- Test cases
- Test assets
- Workflows to manage processes

The integrated test management provides the ability to define test plans, create test cases, associate the test cases with specific test plans, and test the execution scripts to test tools. You can capture the results from the tests for reporting and analysis.

You can enter the defects or changes identified from the test execution or throughout any part of the life cycle into Rational ClearQuest, and the defects and changes will be managed via the workflow process.

Key benefits for SOA projects

Rational ClearQuest provides a number of key benefits for SOA projects:

- Compliance with internal and external requirements using user authentication and authorization and audit trails
- Automated workflow control to enforce processes, which creates repeatable and predictable processes
- Real-time reporting and metrics for providing insight into the processes used
- Integration with Rational ClearCase to define and manage changes to software assets as activities

You can obtain more details about Rational ClearQuest from:

http://www.ibm.com/software/awdtools/changemgmt/index.html

Rational ClearCase

Rational ClearCase V7.0 is a software configuration repository, which provides controlled access to all digital content, includes:

- Requirements
- Design documents
- Models
- Test plans
- Test results

Assets managed under Rational ClearCase can include source code, libraries, documentation, binaries, Web artifacts, or virtually any project artifact that can be digitally represented.

Rational ClearCase supports parallel development support and baseline management with the ability to work independently through private work areas and public integration areas to integrate work into a team environment. Development and integration streams can be separated with capabilities to highlight and accept conflicting changes to ensure the integrity of the baseline assets.

A complete desktop client is provided on Microsoft Windows with remote and Web clients that enable you to access the versioned objects. In addition, integration of Rational ClearCase is provided for a number of leading Integrated Development Environments (IDEs), including Rational Software Development Platform, Eclipse, and Microsoft Visual Studio® 2005. Rational ClearCase is supported with heterogeneous environments and is cross-platform across multiple platforms.

Key benefits for SOA projects

Rational ClearCase provides a number of key benefits to assist you with SOA projects:

- Management and control of assets created during the full SOA life cycle
- Generation of detailed software bill of materials to ensure that an audit trail is maintained of assets and is reproducible
- User authentication and audit trails to assist in meeting compliance requirements for an organization
- Support of the Reusable Asset Specification (RAS) process for the management of package reusable software assets
- Integration with Rational ClearQuest provides the capabilities to define and manage changes to software assets as activities. Using Unified Change Management, a project team or organization can have a complete view of how defects and changes impact assets managed in ClearCase.

You can obtain additional information about Rational ClearCase from:

`http://www.ibm.com/software/awdtools/clearcase/multisite-ext/index.html`

Rational Asset Manager

Rational Asset Manager V7.0 is a development asset management repository to enable organizations to identify, manage, and govern the design, development, and consumption of software assets. It creates a central team repository and processes for the searching and finding of approved assets in an organization. Assets stored in Rational Asset Manager will be relevant to development roles, such as technical managers, analysts, architects, developers, and testers.

Rational Asset Manager can manage many types of assets that include but are not limited to:

- Models
- Components
- Services
- Patterns
- Java, Microsoft .NET, and COBOL source and binaries

Rational Asset Manager provides governance, categorization, and access control and measurement usage for these assets. Industry standards, such as the Reusable Asset Specification (RAS) from the Object Management Group, are supported in Rational Asset Manager. This standard assists in describing assets as part of asset-based development complementing Model Driven Architecture by defining asset production, asset consumption, and asset management.

Support for asset management scenarios is achieved in Rational Asset Manager through the management of asset metadata, including:

- Asset name, description, version, and state
- The contained asset artifact name, description, version, and reference (location)
- Data for integrating to other repositories, such as WebSphere Service Registry and Repository

You can define asset types, compliance, and review processes to enforce compliance to an asset type definition ensuring that correct artifacts are submitted on creation. You can determine impacts on services from changes or defects in a source code file with Rational Asset Manager. Rational Asset Manager can help ensure that you can deploy services to the run time or publish into a service registry, such as WebSphere Service Registry and Repository.

Rational Asset Manager includes an embedded WebSphere Application Server V6.1 for the runtime operation and DB2 Universal Database V9.1 with the installation. It provides an interface using either a Web client or support for an Eclipse V3.2.2-based client, such as products from the IBM Rational Software Delivery Platform V7. External integration with development and runtime environments is supported for products, such as:

- Rational ClearQuest for the management of defects, changes, and creation of more complex customizable asset review processes
- Rational ClearCase for the versioning repository for assets
- WebSphere Service Registry and Repository for the management of runtime services

Key benefits for SOA projects

Rational Asset Manager provides a number of key benefits to assist you with SOA projects:

- Encourages reuse:
  - Via search on asset metadata in Rational Asset Manager
  - Via search and reuse of WebSphere Service Registry and Repository deployed services
- Traceability between deployed services and related assets
- Creation of review workflows for asset governance and compliance for asset review boards
- Access controls based on user, role, group, and asset types
- SOA classifications for integration with WebSphere Service Registry and Repository

You can obtain more details about Rational Asset Manager from:

http://www.ibm.com/software/awdtools/ram/

Products for service deployment

This section describes products used during the service deployment life cycle and highlights their key benefits for SOA projects.

WebSphere Process Server

WebSphere Process Server V6.0.2 is the runtime environment for the deployment of business process and mediation applications to the organization. It leverages and is built on existing WebSphere products, such as WebSphere Application Server V6.0.2 and WebSphere Application Server Network Deployment V6.0.2, and includes WebSphere Enterprise Service Bus V6.0.2.

WebSphere Process Server focuses on two aspects of business processes:

- Orchestration

  Implements services as part of processes with end-to-end transactional processes, human-related tasks, and integration of information from disparate applications. The Business Process Execution Language (BPEL) standard is used for process orchestration.

- Mediation

  Implements mediation and integration capabilities incorporated with WebSphere Enterprise Service Bus. It abstracts the complexities of the integration between applications and data sources by mediating between services in the organization, which shields the business from technology changes.

Key benefit for SOA projects

WebSphere Process Server provides a standards-based business process execution runtime engine for service orchestration and mediation to help you with SOA projects.

You can obtain more details on WebSphere Process Server from:

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

WebSphere Service Registry and Repository

WebSphere Service Registry and Repository V6.0.2 is a solution for the storage, access, and management of service metadata that is deployed to a runtime environment for an organization.

Services in WebSphere Service Registry and Repository can have annotations associated with their selection, invocation, governance, and reuse for the determination in the use of these services. Support is provided for Ontology Web Language (OWL)-based user-defined semantic classifications with annotation information able to be stored in the form of Microsoft Word, Powerpoint, Visio, and Excel®.

WebSphere Service Registry and Repository can be used for the provision of dynamic runtime integration of services using an Enterprise Service Bus (ESB), runtime hosting services, or service mediation. A unified view of these services can be provided via federation of multiple registries and repositories allowing portability and security across the organization.

The interoperability of services is delivered by WebSphere Service Registry and Repository through the support of standards that include:

- Web Services Description Language (WSDL)
- eXtensible Markup Language (XML)
- XML Schema Definition (XSD)
- Business Process Execution Language (BPEL)
- Service Component Architecture (SCA)
- Other standards, such as Universal Description, Discovery, and Integration (UDDI)

Key benefits for SOA projects

WebSphere Service Registry and Repository provides a number of key benefits to assist you with SOA projects:

- Provides SOA life cycle management and governance via:
  - Identification of service associations and relationships
  - Support of discovery and reuse of services with associated metadata and artifacts
  - Enforcement of best practices by instituting standards and best practices

You can obtain additional details about WebSphere Service Registry and Repository from:

http://www.ibm.com/software/integration/wsrr/index.html

WebSphere Business Services Fabric

WebSphere Business Services Fabric V6.0.2 provides an SOA platform for the provision of business services to organizations. It provides the ability to model, assemble, deploy, manage, and govern business services with design time tooling, a runtime environment, and optional prebuilt industry SOA content. WebSphere Business Services Fabric increases the focus in the use of business services for organizations by allowing them to provide:

- Policy-driven services based on:
  - The context in which the service is invoked based on, for example, the type of consumer or the time of innovation
  - The content provided in the service based on certain information in the service message
  - The contract to differentiate the service based on, for example, availability of the service or performance criteria
- Flexible and responsive Business Process Management (BPM) via dynamic service selection and delivery
- Optional industry packs with prebuilt SOA assets, reducing the time to market
- Governance controls to monitor service invocations, performance, and the service life cycle with versions and changes to the services

WebSphere Business Services Fabric is delivered as an integrated product offering that incorporates two major components:

- Business Services Tools Pack (design time), consisting of:
  - WebSphere Integration Developer for orchestrating processes and construction of mediations
  - Composition Studio for visual modeling and management of business services' metadata models and policies
- Business Services Foundation Pack (run time and manage time) consisting of:
  - WebSphere Process Server used for integration and automation of enterprise business processes

  WebSphere Business Services Fabric components:

  - Dynamic Assembler - Highly scalable, service personalization and semantic services engine
  - Business Service Repository - Maintains business service metadata using WebSphere Service Registry and Repository
  - Business Services Subscriber Manager - Provides controls and automates entitlements
  - Business Services Performance Manager - Provides monitoring and visibility of services
  - Business Service Governance Manager - Manages life cycle changes to Business Services metadata

IBM Telecom Operation Pack

The IBM Telecom Operation Pack for WebSphere Business Services Fabric V6.0.2 is an optional feature that contains prebuilt industry content for integration with the WebSphere Business Services Fabric platform. The Telecom Operations Pack provides support for billing, fulfilment, and assurance standards based on telecommunication standards, such as New Generation Operations Systems and Software (NGOSS).

The bundle includes specifically designed assets for the telecom industry that include:

- Prebuilt reference business services templates with associated Web services and metadata using telecom standards:
  - Telecom Applications Map (TAM)
  - NGOSS-enhanced Telecom Operations Map (eTOM)
- Service interfaces with schemas and Web service interfaces for the support of Operational Support System (OSS) and Billing Support System (BSS)

- Operations business glossary and object model using the NGOSS Shared Information Data (SID) model
- Knowledge assets, such as reference architecture and reference guides on installation and development

Key benefits for SOA projects

WebSphere Business Services Fabric provides a number of key benefits to assist you with your SOA projects:

- Governance of run time business services by change management, access rights, and approval processes
- Reuse of services across SOA applications
- Flexible SOA-enabled business process modeling via policy-driven dynamic selection of service endpoints
- Telecom Operations Pack provides a framework for industry specific SOA solutions using telecom standards

You can obtain more details about WebSphere Business Services Fabric, including the industry packs, from:

http://www.ibm.com/software/integration/wbsf/index.html

Products for service management

This section describes products used during the service management life cycle and highlights their key benefits for SOA projects.

Tivoli Change and Configuration Management Database

Tivoli Change and Configuration Management Database V1.0 is an enterprise-wide configuration management database and platform to discover and federate IT information spread across the enterprise. Information includes details about servers, storage devices, networks, middleware, applications, and data.

Tivoli Change and Configuration Management Database provides the platform for creating IT Infrastructure Library (ITIL) best practices functions in an organization's environment. Some of the key functional IT infrastructure capabilities:

- Change and configuration

  Access key information about configuration items and their relationship, with the ability to identify business impacts and see all pending change and release schedules.

- Advanced mapping

  Automated discovery to automate the creation and maintenance of application infrastructure with complete runtime dependencies, in-depth configuration values, and accurate change history. Support is provided for:

  – IBM AIX®, Linux®, Microsoft Windows, HP-UX, Sun Solaris™, and IBM z/OS®

  – Web servers, application servers, databases, and middleware software

  – Network elements, such as routers, switches, and load balancers

  – Storage elements, such as host bus adapters (HBAs), storage area networks (SANs), switches, and disk arrays

  – Packaged applications, such as SAP®

- Automation of process workflows

  Change and configuration management include process workflows integrating with IBM Tivoli Process Management with the ability to create customized processes for an organization's processes. Analysis is provided of prior activities and tasks to assist in decision making with the capability to receive confirmation of any success change or deployment.

- Enforcing policies for compliance

  Ability to assess system configuration compliance against predefined and authorized baselines and to integrate policies. For example, security privileges of configuration files can be assessed to ensure that they are only assessed by authorized users.

Key benefits for SOA projects

Tivoli Change and Configuration Management Database provides a number of key benefits to assist you with your SOA projects:

- Management of change and configuration of all relevant assets
- Creation and monitoring of policies around assets
- Auditability of change and assess of assets

Other Tivoli products

This section provides a summary of the IBM software products that are used to manage services and supporting infrastructure.

> **Note:** These products are identified for information only and will not be applied in the context of this book. Further information is provided about each product.

IBM Tivoli Monitoring (ITM) products

IBM Tivoli Monitoring (ITM) is the primary product that provides the base infrastructure for management and monitoring. The technology for ITM originated from and is shared with OMEGAMON®, which was obtained in the Candle® acquisition by IBM. The OMEGAMON product name is still used on the z/OS platform.

The core components included with IBM Tivoli Monitoring V6.1 are:

- Tivoli Enterprise Monitoring Server (TEMS)

  The Tivoli Enterprise Monitoring Server (TEMS) is the core component of an IBM Tivoli Monitoring solution. TEMS is responsible for collecting alerts, performance, and availability data from agents. In addition, TEMS is used to track the heartbeat request interval for all configured TEMS agents. TEMS initiates and tracks all situations and policies and stores the data in the centralized data warehouse.

  When installing TEMS, the primary Monitoring Server is configured as a *hub*, and all subsequent Monitoring Servers are configured as remote to the Hub. This type of architecture provides for great scalability and centralized collection and analysis of the data.

- Tivoli Enterprise Portal Server (TEPS)

  The Tivoli Enterprise Portal Server (TEPS) provides a presentation layer and database repository for all graphical presentation of monitoring data. The TEPS is used for retrieval, manipulation, analysis, and formatting of data. It manages this access through user workspace views. The TEPS maintains a persistent connection to the TEMS Hub and can be considered a logical gateway between the TEMS Hub and the TEP client (desktop or browser).

  The TEPS provides the ability to customize workspace views, situations (thresholds), and workflows. A database manager, such as DB2 Universal Database, is used to host the Data Warehouse and TEPS configuration database.

The following products have integrated interfaces into the Tivoli Enterprise Portal to provide a consolidated view of composite application data:

- IBM Tivoli Composite Application Manager for SOA
- IBM Tivoli Composite Application Manager for WebSphere
- IBM Tivoli Composite Application Manager for Response Time Tracking
- IBM OMEGAMON XE product family
- IBM Tivoli Monitoring product family
- IBM Tivoli Enterprise Console® (TEC)
- IBM NetView® for z/OS (release 5.2)

- Tivoli Enterprise Portal (TEP) clients

  The Tivoli Enterprise Portal can be accessed using the following clients to view all monitoring data collection within a single window:

  - TEP Desktop client: Java application
  - TEP Browser client: Java applet run in a Web browser

- Tivoli Enterprise Monitoring Agents

  The Tivoli Enterprise Monitoring Agents are installed on the systems or subsystems requiring data collection and monitoring. The agents are responsible for data gathering and distribution of attributes to the monitoring servers, including initiating the heartbeat status.

IBM Tivoli Composite Application Manager products

The IBM Tivoli Composite Application Manager (ITCAM) product family consists of the following products:

- ITCAM for SOA: Monitor and manage the SOA services layer
- ITCAM for Response Time Tracking: Proactively recognize, isolate, and resolve transaction performance problems
- ITCAM for WebSphere: Isolate the root cause of bottlenecks in a WebSphere application runtime environment
- ITCAM for CICS Transactions: Capture data from CICS systems to be analyzed in ITCAM for Response Time Tracking and WebSphere
- ITCAM for IMS Transactions: Capture data for IMS systems to be analyzed in ITCAM for Response Time Tracking

IBM Tivoli OMEGAMON XE products

There are many IBM Tivoli OMEGAMON XE products. Two key products used within the SOA space are:

- Tivoli OMEGAMON XE for Messaging: Monitor MQ family run times and provide automatic corrective actions to improve performance and availability
- Tivoli OMEGAMON XE for CICS: Monitor and manage CICS transactions and resources so that, when a problem occurs, it can be quickly detected and isolated to minimize or eliminate any impact to your business

IBM Tivoli Service Level Advisor

IBM Tivoli Service Level Advisor (TSLA) is a Service Level Management solution for providers of IT services. It simplifies and automates the process of managing service level agreements, enabling IT organizations to proactively manage and report on service levels from across the management infrastructure.

IBM TSLA is a predictive solution for defining, analyzing, and reporting on SLAs enterprise-wide. Among the key features are the ability to:

- Leverage wizards to rapidly define SLAs
- Automate SLA evaluation by enabling alerts for violations
- Provide executive-level reports to effectively communicate SLA performance to executives across the business
- Avoid violations by analyzing trends
- Be integrated with IBM Tivoli Monitoring. This integration provides the ability to create TSLA reports based on data (events) mined from the TM data warehouse database. The reports can be a very useful tool to determine if SLAs are being achieved to ensure SOA governance.

Where to obtain more information

You can obtain more information about IBM Tivoli Monitoring, ITCAM products, and IBM Tivoli Service Level Advisor at:

- IBM Tivoli management products Information Centers:

  http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp?topic=/com.ibm.tivoli.az.doc/welcome.htm
- *IBM Tivoli Composite Application Manager V6.1 Family Installation, Configuration, and Basic Usage*, SG24-7151
- *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155

IBM Tivoli Unified Process Composer

IBM Tivoli Unified Process Composer is based on industry best practices and enables you to easily understand processes, the relationships between processes, and the roles and tools involved in an efficient process implementation. The IBM Tivoli Unified Process Composer complements the ITUP content with IBM Rational Method Composer, which enables you to adopt ITUP processes as shipped or to use ITUP elements as building blocks in processes that are tailored to the needs of your organization.

IBM Tivoli Unified Process (ITUP) Composer V2.1 provides detailed documentation of IT Service Management processes based on industry best practices, enabling users to significantly improve their organization's efficiency and effectiveness. ITUP Composer is the product version of the free IBM Tivoli Unified Process tool. ITUP Composer provides more detailed content and tooling to enable content customization, extension, and publishing.

You can obtain more information about IBM Tivoli Unified Process Composer from:

http://www.ibm.com/software/tivoli/products/unified-process-composer/

The method content is available from:

http://rchgsa.ibm.com/home/j/1/j1long/web/public/ITUP%20Project/ITUP%20Composer/itupc2-1/index.htm

B

WebSphere Business Services Fabric installation

This appendix describes how to install WebSphere Business Services Fabric and the related Telecom Operation Pack.

It contains the following sections:

- Supporting products required
- Building the architecture for run time

Supporting products required

In this book, we use Windows 2003 Server as the operating system. This section describes the products that we need to support WebSphere Business Services Fabric.

WebSphere Process Server V6.0.2

For more information about installing this product, see:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wsps.602.ins.doc/welcome_wps_ins.html

Then, update it in order to reach 6.0.2.2 level. For more details, see:

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

WebSphere Integration Developer V6.0.2

For more information about installing this product, see:

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wbit.help.nav.doc/topics/welcome.html

Then, update it in order to reach 6.0.2.2ifix001-20070801_0929 level. For more details, see:

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

Install WebSphere Business Services Fabric Tool Pack for WebSphere Integration Developer in order to add plug-ins for the Business Service Composition Studio.

DB2 Universal Database Enterprise Edition V8

For more information about installing this product, see:

http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp

Then, update it in order to reach V8.2.5 level. For more details, see:

http://www-306.ibm.com/software/data/db2/support/db2_9/

WebSphere Service Registry and Repository V6.0.2

For more information about installing this product, see:

http://publib.boulder.ibm.com/infocenter/sr/v6r0m2/index.jsp?topic=/com.ibm.sr.doc/twsr_installn02.html

Building the architecture for run time

Now that all of the products that we need are identified and the environment is prepared, we must build the environment.

This is described in the following sections:

- Creating a WebSphere Process Server BPC-enabled profile

  First, create a new WebSphere Process Server profile in order to prepare it for the installation of WebSphere Business Services Fabric on it. A new profile is required, because the default profile (provided with the installation of WebSphere Process Server) does not support Business Process Choreography (BPC).

- Installing WebSphere Business Services Fabric

  We install WebSphere Business Services Fabric into the newly created WebSphere Process Server profile. This includes the installation of the Foundation Pack and the Telecom Operation Pack.

Creating a WebSphere Process Server BPC-enabled profile

Because business process choreography is necessary for running this process, we first must create a profile that supports BPC. Therefore, we create a new profile using the Profile Creation Wizard provided by WebSphere Process Server:

1. From the installation directory of WebSphere Process Server (in this example, it is `E:\Programs\WebSphere\ProcServer\`), run bin\ProfileCreator_wbi\pcatWindows.exe.
2. The Welcome page appears. Click **Next**.
3. Specify the profile name to be used. In this case, we use `ProcServerBPC` as the profile name. Check **Make this profile the default** as in Figure B-1 on page 304. Click **Next.**

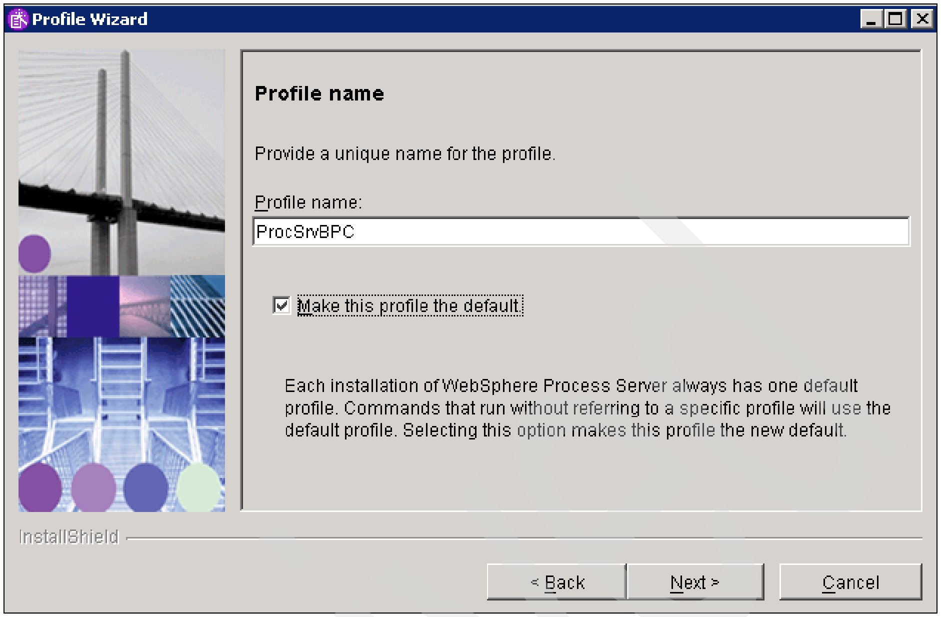


Figure B-1 Profile name

4. Enter or click **Browse** to specify the directory where the profile will be stored. In this example, we use a non-default directory: `E:\programs\WebSphere\ProcServer\profiles\ProcSrvBPC` as in Figure B-2 on page 305. Click **Next**.

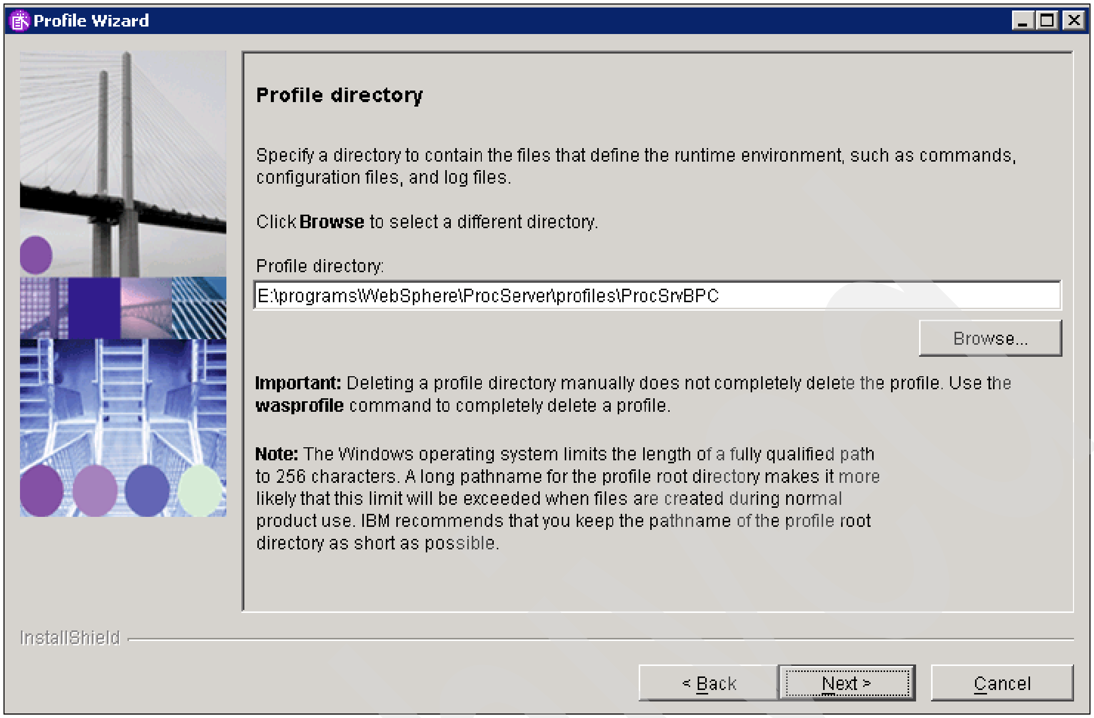


Figure B-2 Profile directory

5. Specify host and node names for the new profile as in Figure B-3 on page 306. Click **Next**.

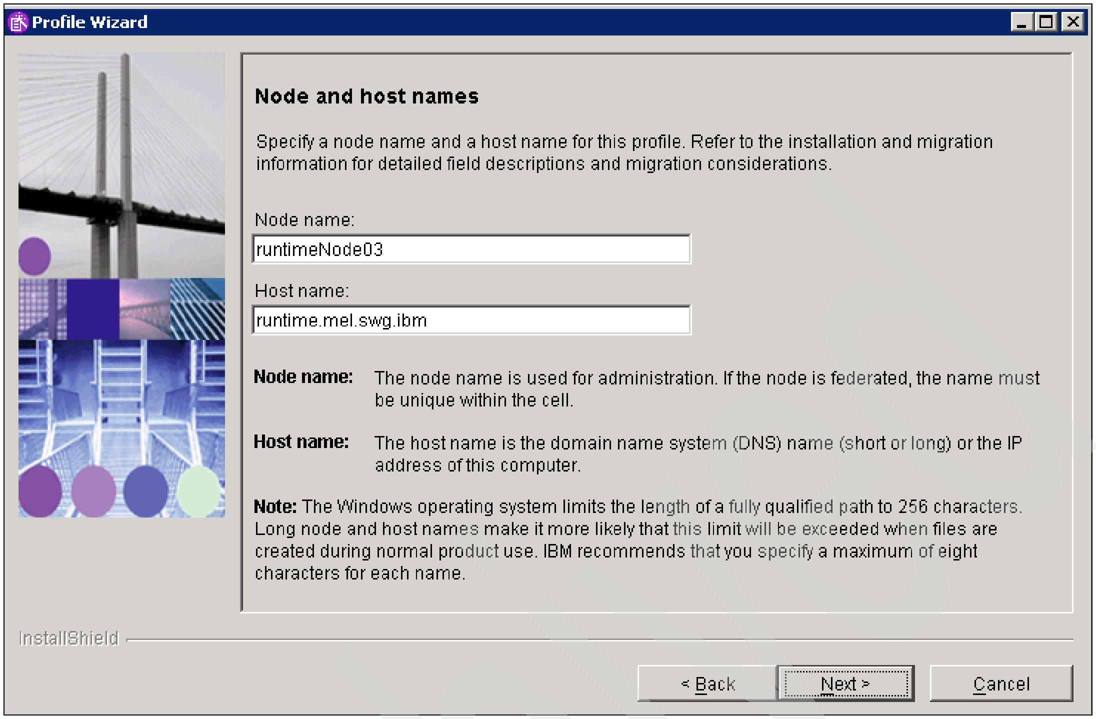


Figure B-3 Node and host names

6. Specify ports for this profile. In this example, we have kept the default ports as in Figure B-4 on page 307. Click **Next**.

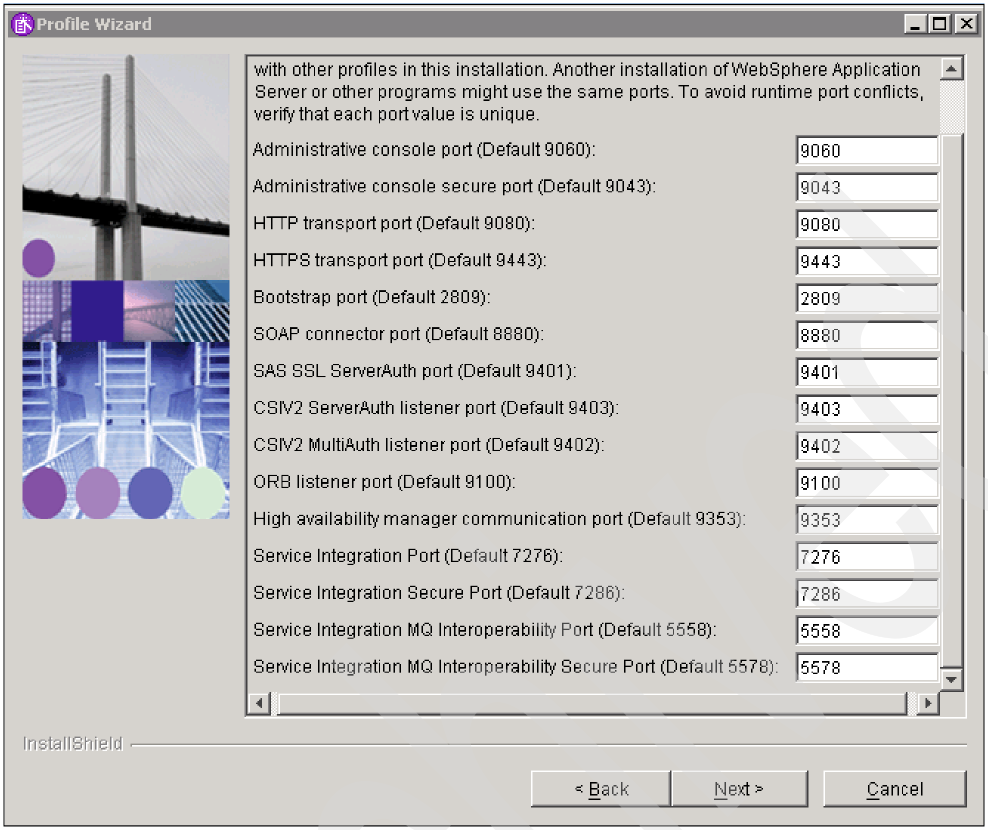


Figure B-4 Ports for WebSphere Process Server new profile

7. Specify if you want this profile to run as a Windows service. In this example, we have disabled it. Click **Next**.
8. Specify the Service Components Architecture configuration. In this example, we cleared the **Configure the Service Integration Bus in a secured mode** box in order to not use security for this configuration. Click **Next.**
9. Configure the Common Event Infrastructure. Enter user `wmqmgr` with password `wmqmgr` for WebSphere Messaging queue manager, select `server1` as the WebSphere server and use `Cloudscape V5.1` as the database provider for this database, as in Figure B-5 on page 308. Oracle® is also supported as another database. In this example, you choose between them.

   Create new database objects on an existing instance (OCI) or generate scripts to create it.

   In both cases, you must provide information for the database instance, the user name, and the password to create in the database (the user and the password must have SYSDBA authority), and the JDBC driver location.

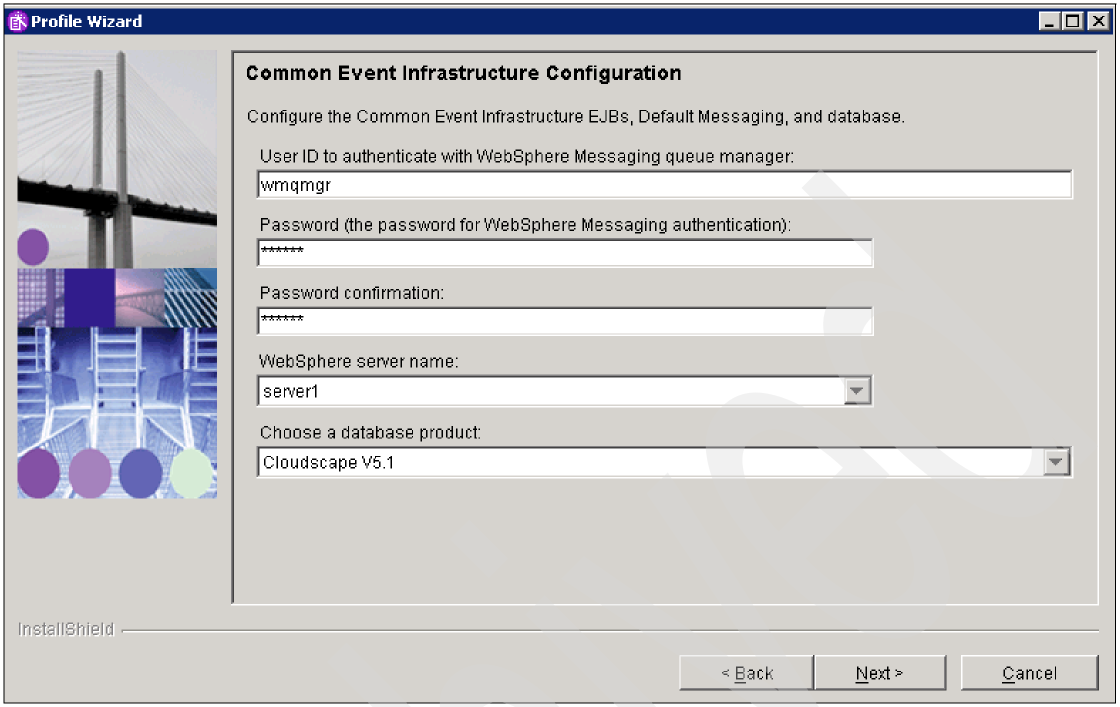


Figure B-5 Common Event Infrastructure Configuration

10. Check **Configure a sample Business Process Choreographer** in order to specify that this profile will support BPC. Also, enter user `sibuser` and password `sibuser` for connecting to Service Integration Bus as in Figure B-6 on page 309.

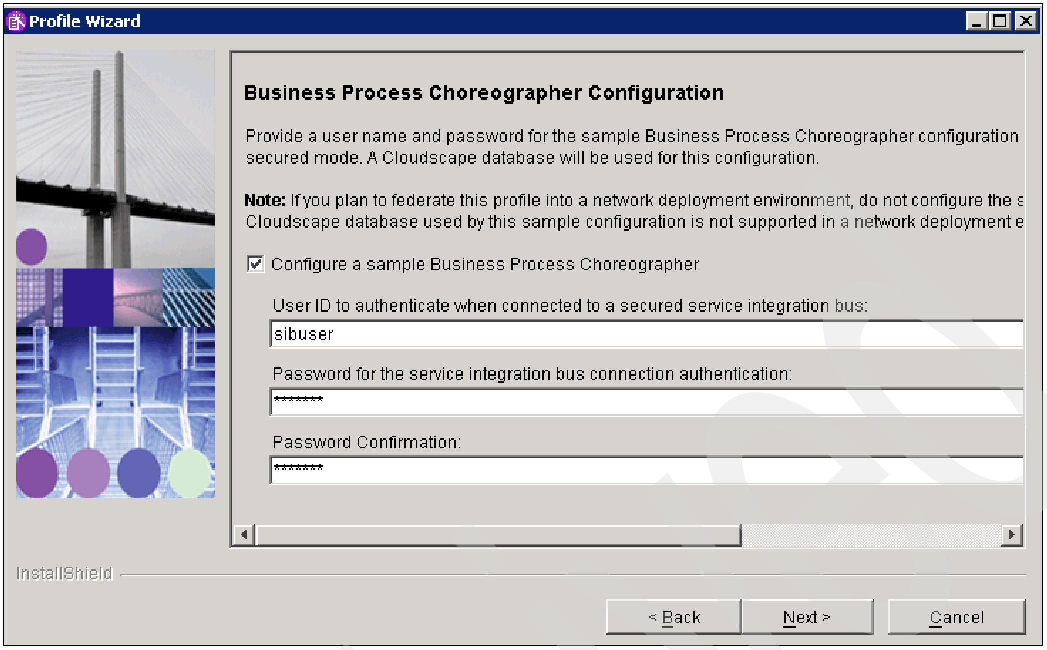


Figure B-6 Business Process Choreographer Configuration

11. Select the server to use as the Application Scheduler; by default, this is `server1`. Click **Next**.
12. Finally, specify the database configuration information for this profile. See Figure B-7 on page 310. Many database providers are supported here. If we choose Oracle again, we have to use an existing database or create a new database, so we have to provide information about the user and password, driver location, and type (OCI or Thin), host, and the port of the database to which we want to connect.

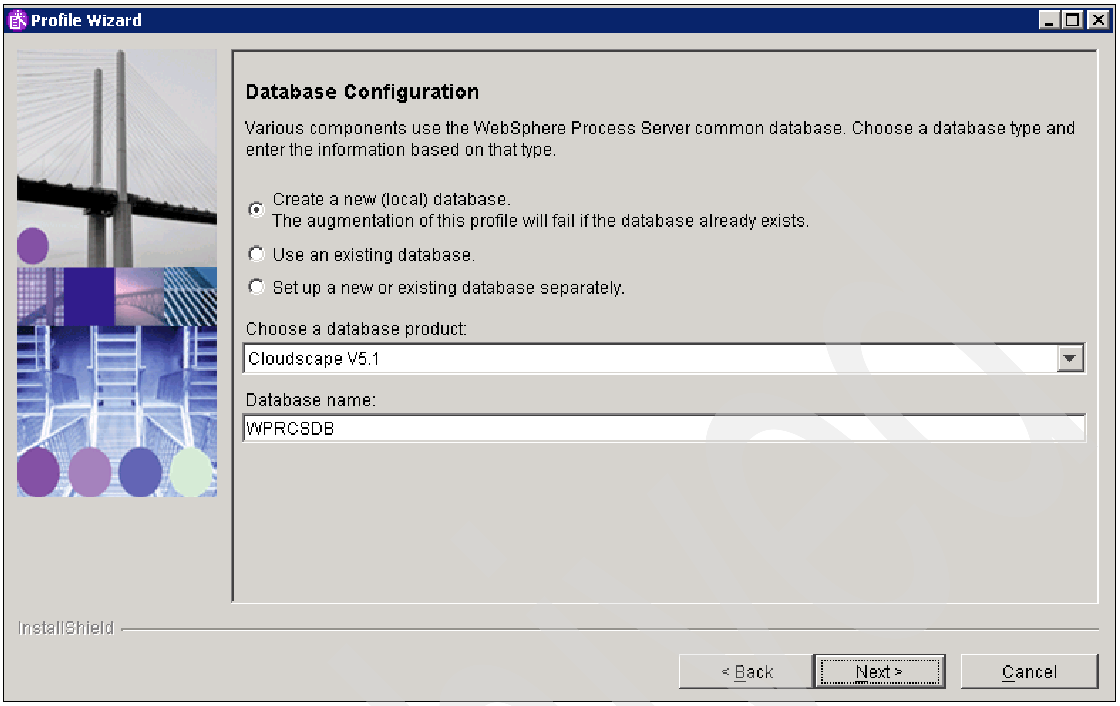


Figure B-7 Database Configuration

Installing WebSphere Business Services Fabric

This product has one Foundation Pack, which has the basic tools, engine, and catalog for WebSphere Business Services Fabric tools, and specialized industry packs in order to focus on each industry semantic, such as telecommunications, health, banking, and insurance.

Installing the Foundation Pack

After we have installed WebSphere Process Server and created its Business Process Choreography (BPC)-enabled profile, we are now ready to install WebSphere Business Services Fabric Foundation Pack:

1. From the installation directory where you downloaded WebSphere Business Services Fabric Foundation Pack, run `installers\install_fabric_win.exe`. This will bring up the Installation Welcome panel. Click **Next**.
2. Select all of the packages to be installed as in Figure B-8 on page 311. Click **Next**.

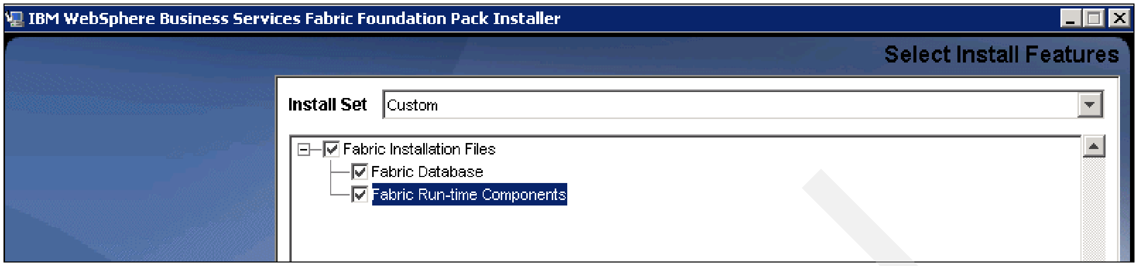


Figure B-8 WebSphere Business Services Fabric package selection panel

3. Select the directory to which to install, for example, `E:\programs\WebSphere\Fabric\FoundationPack` and then click **Next** as in Figure B-9.

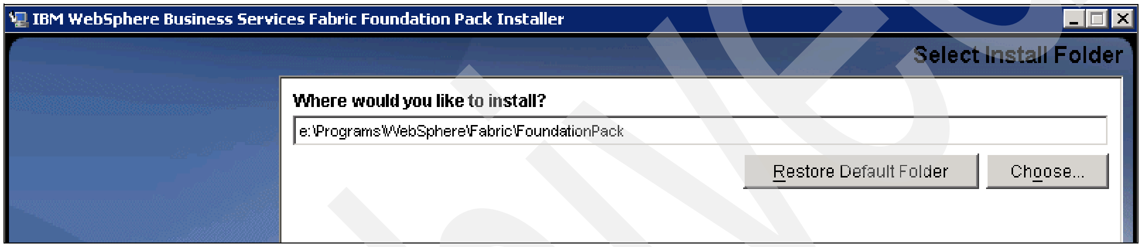


Figure B-9 WebSphere Business Services Fabric installation directory

4. Select the Drive where the database will reside. In this example, select `E:` and then click **Next** as in Figure B-10.

Figure B-10 Drive where database will reside

5. Specify the user name and the password for the database administrator. This will be used to create the WebSphere Business Services Fabric database. In this example, we use `db2admin` as the user name and `db2admin` as the password as shown in Figure B-11 on page 312. Click **Next**.

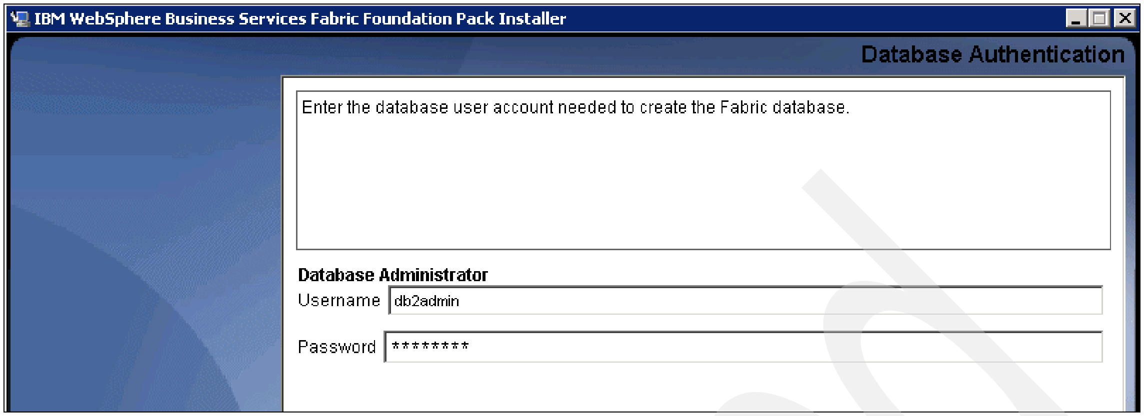


Figure B-11 Database user account information

6. Click **Choose** to specify the WebSphere Process Server installation directory as in Figure B-12. Click **Next**.

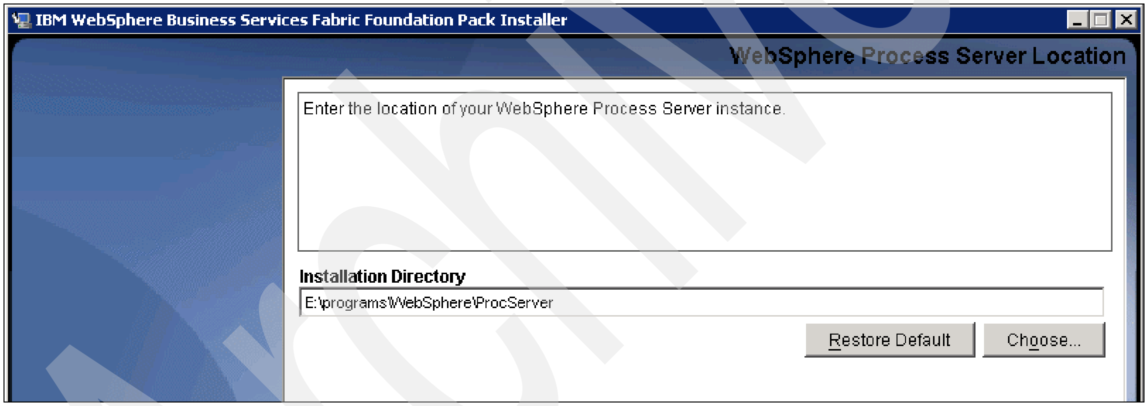


Figure B-12 Process Server installation directory for WebSphere Business Services Fabric

7. Select the WebSphere Process Server profile in which WebSphere Business Services Fabric will be installed. Remember it has to be the same profile created in “Creating a WebSphere Process Server BPC-enabled profile” on page 303, because it has to support BPC. In this example, we use `ProcSrvBPC` as in Figure B-13 on page 313. Click **Next**.

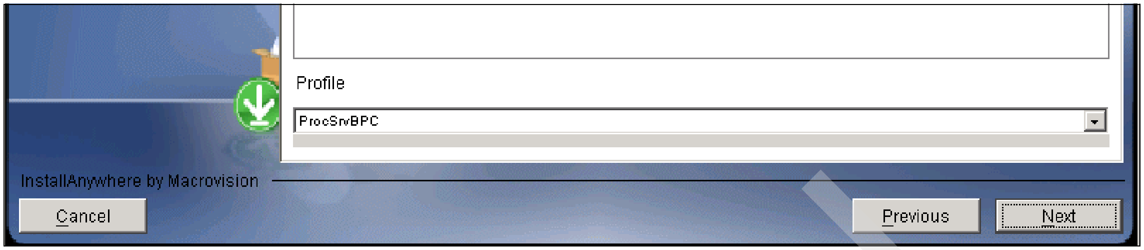


Figure B-13 Process Server profile in which WebSphere Business Services Fabric will be installed

8. Specify which security mechanism WebSphere Business Services Fabric is going to use. In this example, we select the default, which does not use security. Select `None` as in Figure B-14. Click **Next**. In this step, we can choose other alternatives, such as `LDAP` or `Other`, which can be mapped to an authentication service already available in the company.

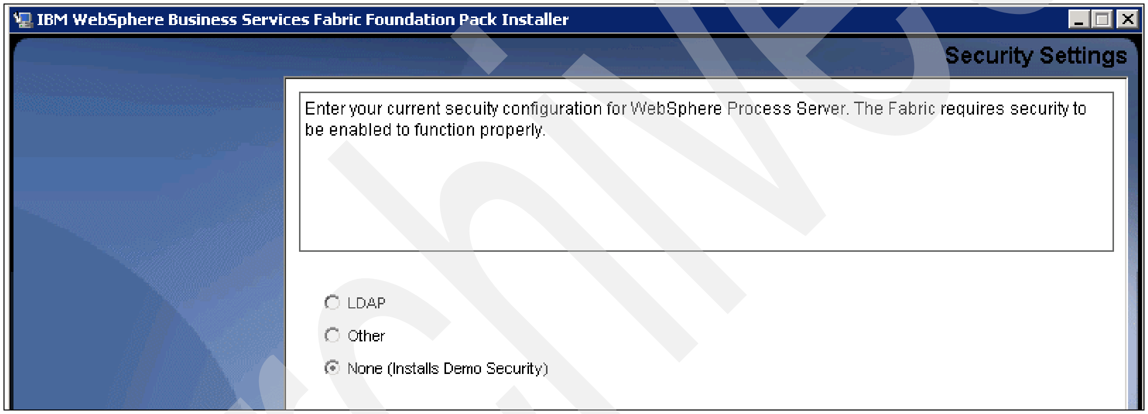


Figure B-14 Security mechanism for WebSphere Business Services Fabric

9. Click **Choose** to specify all JDBC driver settings for WebSphere Business Services Fabric access to the database. In this example, we use the DB2 Java installation directory `E:\programs\db2\SQLLIB\java` as in Figure B-15 on page 314. Click **Next**.

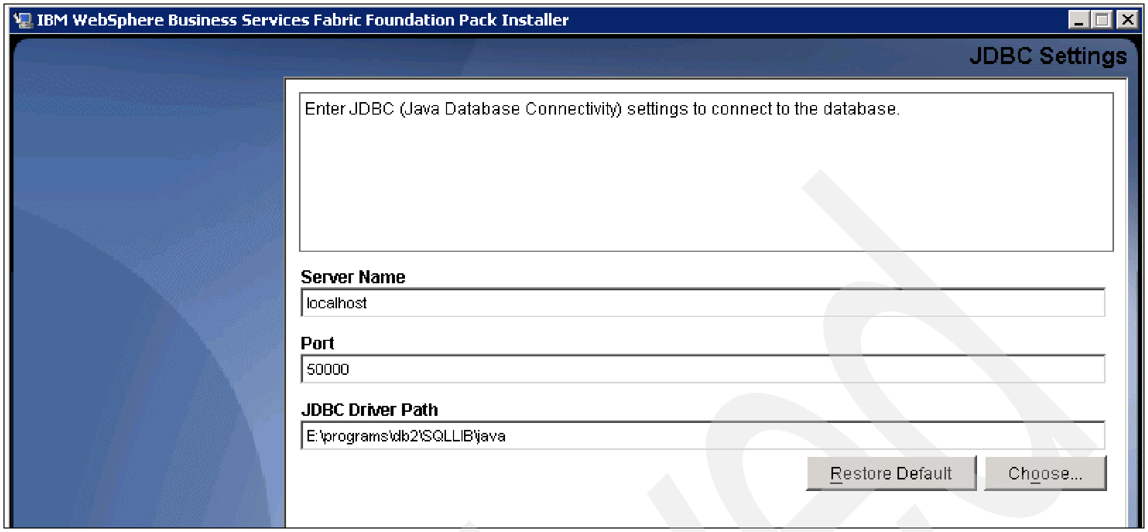


Figure B-15 JDBC Settings details

10. Specify mail details for WebSphere Business Services Fabric to send user notifications as in Figure B-16. Click **Next**.

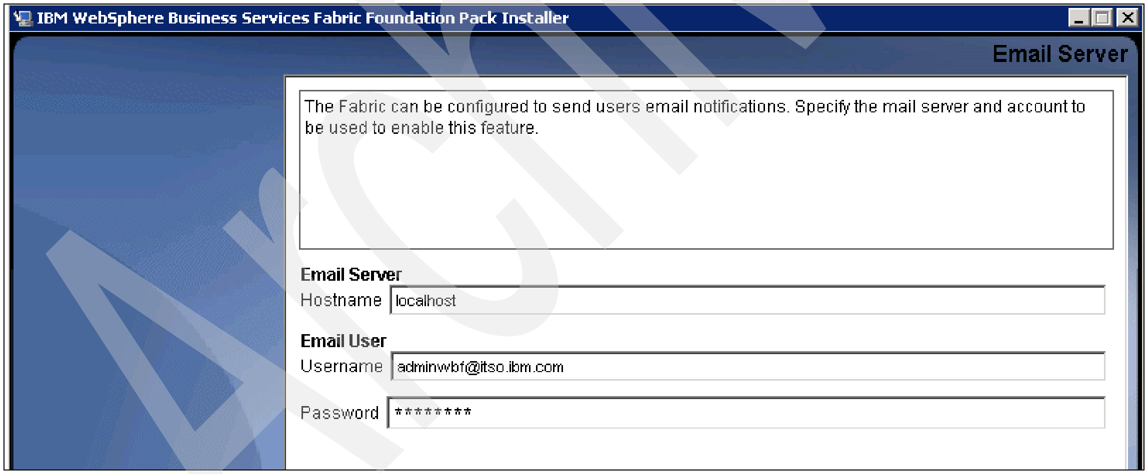


Figure B-16 Mail server and account specification

11. Finally, the summary page is shown. Click **Install** as in Figure B-17 on page 315.

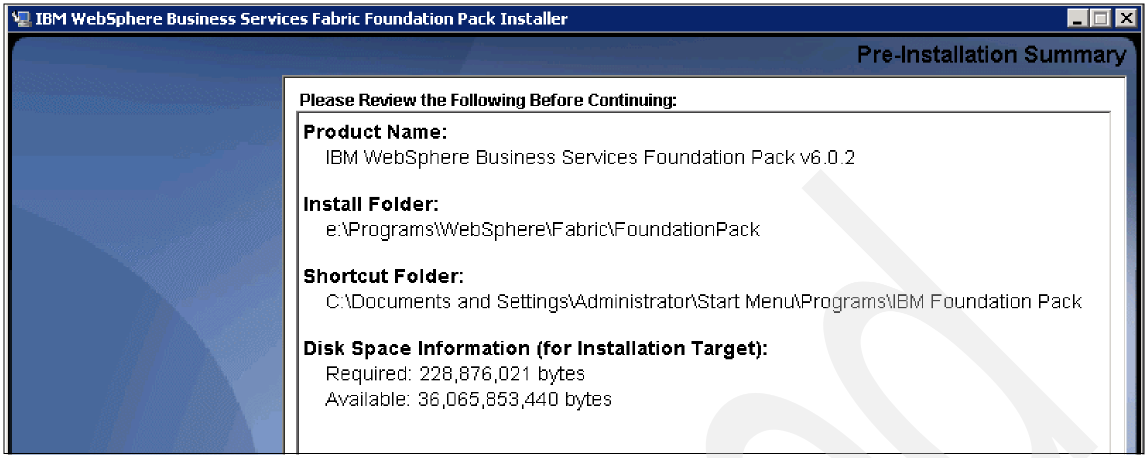


Figure B-17 Installation of WebSphere Business Services Fabric is ready to start

Verifying the installation

After installation, new enterprise applications need to reside in the WebSphere Process Server profile environment. We used default ports so that we can access the WebSphere Business Services Fabric console using the URL:

`http//localhost:9080/fabric/app`

This is shown in Figure B-18 on page 316.

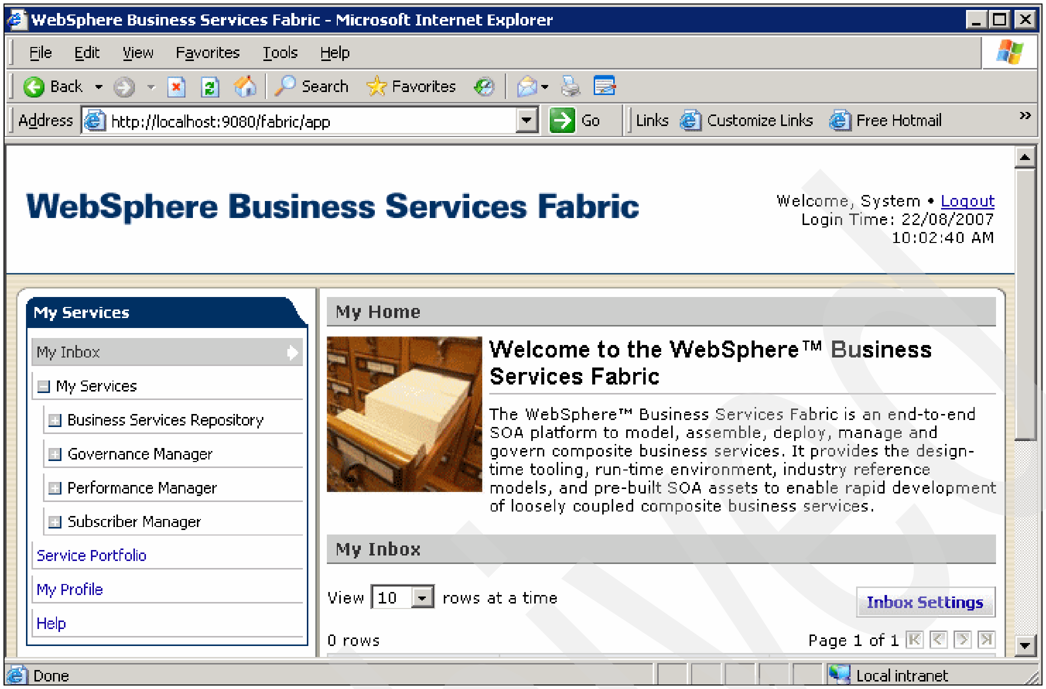


Figure B-18 WebSphere Business Services Fabric console

Installing the Telecom Operation Pack

After we have installed the WebSphere Business Services Fabric Foundation Pack, we can install specific industries packs. For this book, we are interested in the Telecom Operation Package.

To install it, perform these steps:

1. From the installation directory to where WebSphere Business Services Fabric Telecom Operation Package was downloaded, run `installers\install_telecom_win.exe`.
2. You will see the Welcome page. Click **OK**, and you will see the installation page as in Figure B-19 on page 317. Click **Next**.

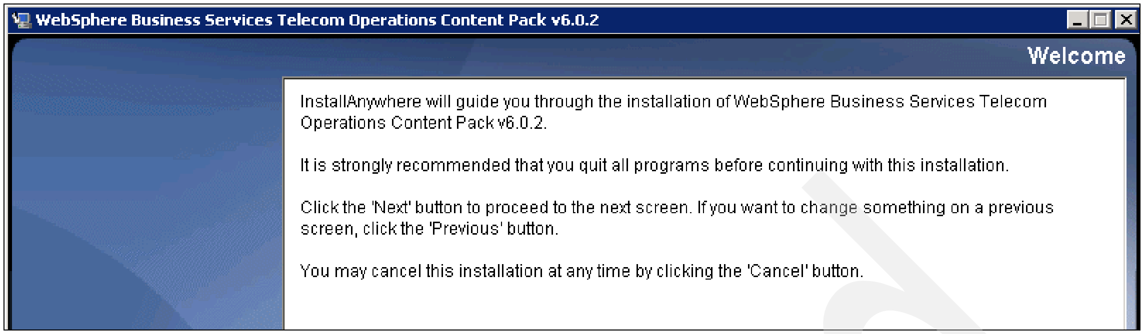


Figure B-19 Installation of Telecom Operation Package

3. Read the license agreement page, select **I accept the terms in the license agreement** and click **Next** (Figure B-20).

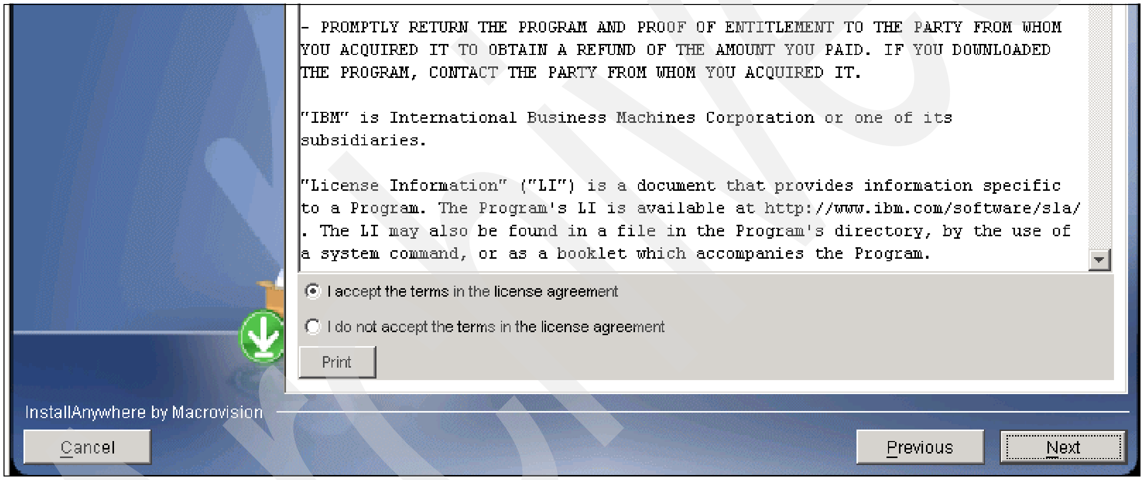


Figure B-20 License agreement

4. Select the packages to be installed. We select both packages as in Figure B-21.

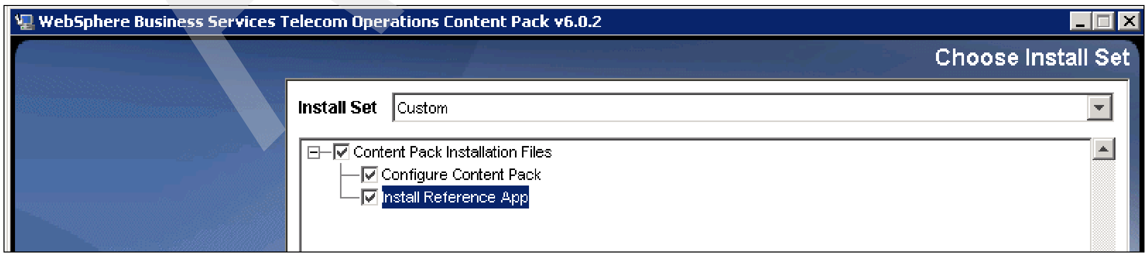


Figure B-21 Telecom Operation Package Installation packages

5. Click **Choose** or enter the installation directory for the Telecom Operation Package, as in Figure B-22. Click **Next**.

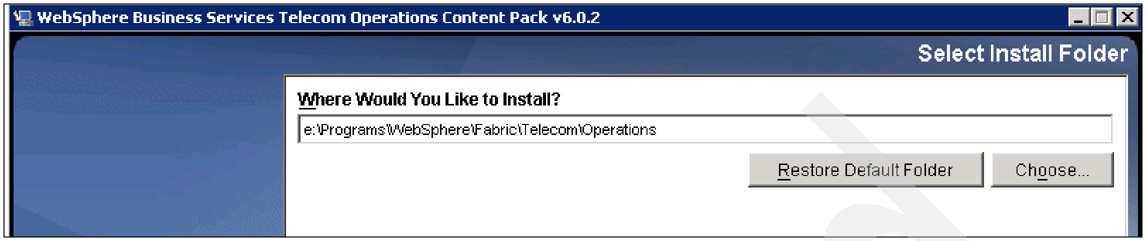


Figure B-22 Telecom Operation Package Installation directory

6. Click **Choose** or select the installation directory where WebSphere Process Server is installed as in Figure B-23. Click **Next**.

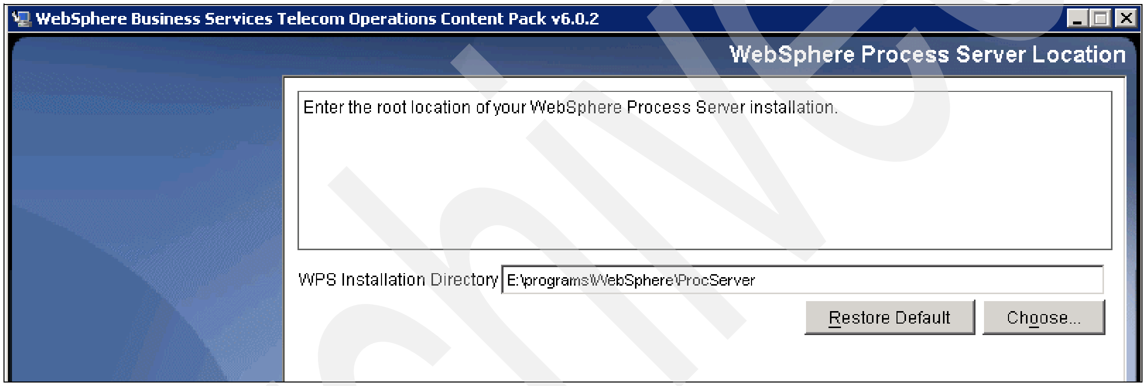


Figure B-23 WebSphere Process Server installation directory

7. Select which WebSphere Process Server profile will be used for this installation. The same process that we follow for WebSphere Business Services Fabric Foundation Pack profile selection must be used in this step, as in Figure B-24. Click **Next**.

Figure B-24 WebSphere Process Server profile selection

8. Specify the server location where WebSphere Business Services Fabric is listening. We installed it using the default port location, so the values must be `localhost` as the host name and `9080` as the port name. The user ID is `admin` and password is `webify`. See Figure B-25. Click **Next**.

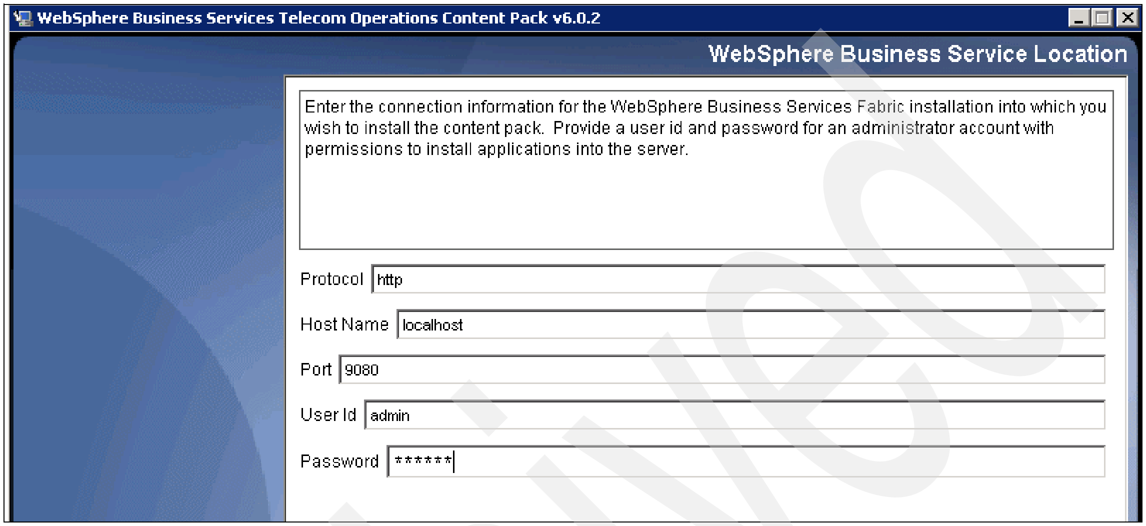


Figure B-25 WebSphere Business Services Fabric server location

9. Finally, everything is ready for installation, as in Figure B-26. Click **Install**.

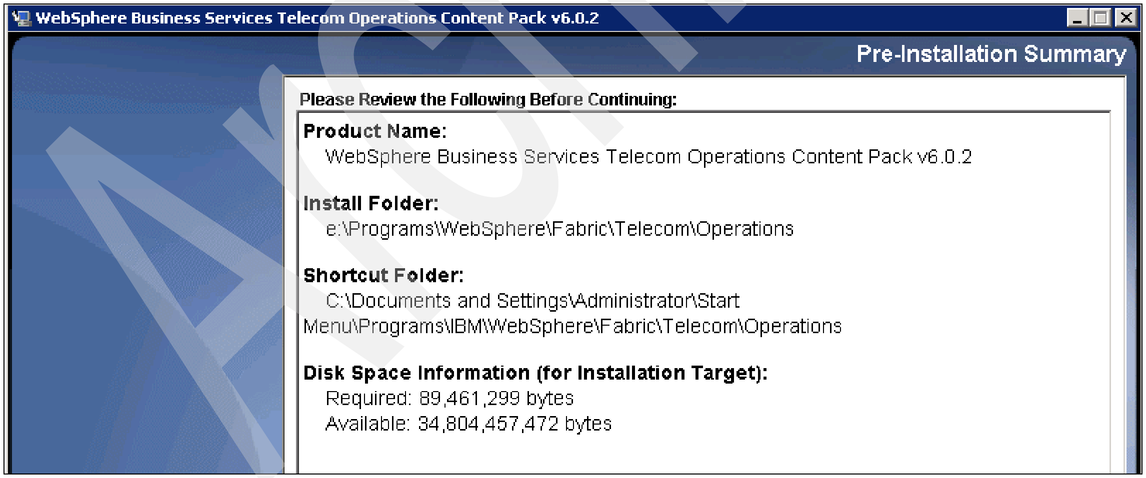


Figure B-26 Telecom Operation Package installation is ready to start

Abbreviations and acronyms

ABEs	Aggregate Business Entities
ARTS	Association for Retail Technology Standards
BPEL	Business Process Execution Language
BPM	Business Process Management
BSM	Business service management
BSS	Billing Support System
CCMDB	Change and Configuration Management Database
CMDB	Configuration Management Database
CML	Common Model Library
CMMI	Capability Maturity Model Integration
CMP	Container-Managed Persistence
CoE	Center of Excellence
CRM	Customer Relationship Management
CSF	Critical Success Factors
CSI	Continual Service Improvement
CSV	Comma-separated values
DXSI	DataXtend Semantic Integrator
EAR	enterprise archive
ESB	Enterprise Service Bus
eTOM	enhanced Telecom Operations Map
FDL	Flow Definition Language
HBA	host bus adapter
IBM	International Business Machines Corporation
IDE	Integrated Development Environment
IE	Information Engineering
ISO	International Organization for Standardization
IT	Information Technology
ITGI	IT Governance Institute
ITIL	IT Infrastructure Library
ITSM	IT Service Management
ITSO	International Technical Support Organization
ITUP	IBM Tivoli Unified Process
J2EE	Java Enterprise Edition
JSF	JavaServer Faces
KGI	Key Goal Indicators
KPI	Key Performance Indicators
KPIs	Key Performance Indicators
LDMs	Logical Data Models
MAC	Media Access Control
MDA	Model Driven Architecture
MDRs	Management Data Repositories
NGOSS	New Generation Operations Systems and Software
OGC	Office of Governance Commerce
OMG	Object Management Group
OSIMM	Open Group Services Integration Maturity Model
OSS	Operational Support System
OWL	Ontology Web Language
QoS	Quality of Service

RAS	Reusable Asset Specification
RUP	Rational Unified Process
SAN	storage area networks
SCA	Service Component Architecture
SCDL	Service Component Description Language
SCOR	Supply-Chain Operations Reference
SD	Service Design
SE	Standard Edition
SGMM	SOA Governance and Management Method
SID	Shared Information Data
SLA	Service Level Agreement
SML	Service Modeling Language
SO	Service Operation
SOA	Service-oriented architecture
SS	Service Strategy
ST	Service Transition
TAM	Telecom Applications Map
TEC	Tivoli Enterprise Console
UML	Unified Model Language
WAR	Web archive
WBS	Work Breakdown Structure
WSDL	Web Services Description Language
WSIL	Web Services Inspection 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 book.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 324. Note that some of the documents referenced here may be available in softcopy only:

- *A Practical Guide to the IBM Autonomic Computing Toolkit*, SG24-6635
- *Best Practices for SOA Management*, REDP-4233
- *Building SOA Solutions Using the Rational SDP*, SG24-7356
- *CICSPlex SM Business Application Services: A New Solution to CICS Resource Management*, SG24-5267
- *Composite Application Provisioning with IBM Tivoli Provisioning Manager V3.1*, REDP-4222
- *Deployment Guide Series: IBM Tivoli Change and Configuration Management Database Configuration Discovery and Tracking v1.1*, SG24-7264
- *Deployment Guide Series: IBM Tivoli Provisioning Manager Version 5.1*, SG24-7261
- *End-to-end Automation with IBM Tivoli System Automation for Multiplatforms*, SG24-7117
- *IBM Tivoli Business Service Manager V4.1*, REDP-4288
- *IBM Tivoli Composite Application Manager V6.1 Family Installation, Configuration, and Basic Usage*, SG24-7151
- *IBM Tivoli OMEGAMON XE V3.1.0 Deep Dive on z/OS*, SG24-7155
- *Large-Scale Implementation of IBM Tivoli Composite Application Manager*, REDP-4162
- *Patterns: SOA Foundation - Business Process Management Scenario*, SG24-7234
- *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240

- *Solution Deployment Guide for IBM Tivoli Composite Application Manager for WebSphere*, SG24-7293
- *The IBM Rational Unified Process for System z*, SG24-7362
- *Using a Single Business Pattern with the Rational Unified Process (RUP)*, REDP-3877
- *z/OS Technical Overview: WebSphere Process Server and WebSphere Enterprise Service Bus*, REDP-4196

Other publications

These publications are also relevant as additional information sources:

- *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results,* Peter Weill and Jeanne Ross. ISBN 1591392535
- *Getting Started with the SID* book by John Reilly of the TM Forum. The book is published by TM Forum and is available at:

  http://www.tmforum.org/Bookstore/GettingStartedWith/33446/article.html

How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks publications, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy IBM Redbooks publications, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

IBM

Redbooks

Implementing Technology to Support SOA Governance and Management

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

IBM®

Implementing Technology to Support SOA Governance and Management

Redbooks®

Guidelines for establishing SOA governance and management

Service lifecycle management with Rational Asset Manager

Runtime discovery with WebSphere Business Services Fabric

Implementing SOA governance and management requires the consideration of three pillars: people, process, and technology. This IBM® Redbooks® publication addresses the third pillar, technology, demonstrating how to build advanced SOA solutions, and is intended for Technology professionals, Program Managers, Business Analysts, and Business Sponsors, including C-level executives.

The first part introduces SOA governance and management, defines the IBM SOA governance and management method of plan, define, enable, and measure, and addresses the IBM SOA Foundation, SOA metadata, and the service life cycle.

The second part uses a fictional telecommunications company business scenario to illustrate the use of technology in implementing SOA solutions with governance and management. This includes a detailed hands-on introduction to Rational Asset Manager, highlighting its governance capabilities. An additional chapter discusses the use of WebSphere® Business Services Fabric as a process run time and WebSphere Service Registry and Repository as a runtime repository for service discovery.

We discuss how a layered mediation architecture supports SOA governance and how DataXtend Semantic Integrator can be used to implement service components for data mediation. Finally we discuss additional solutions for SOA management.

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

SG24-7538-00 ISBN 0738485535