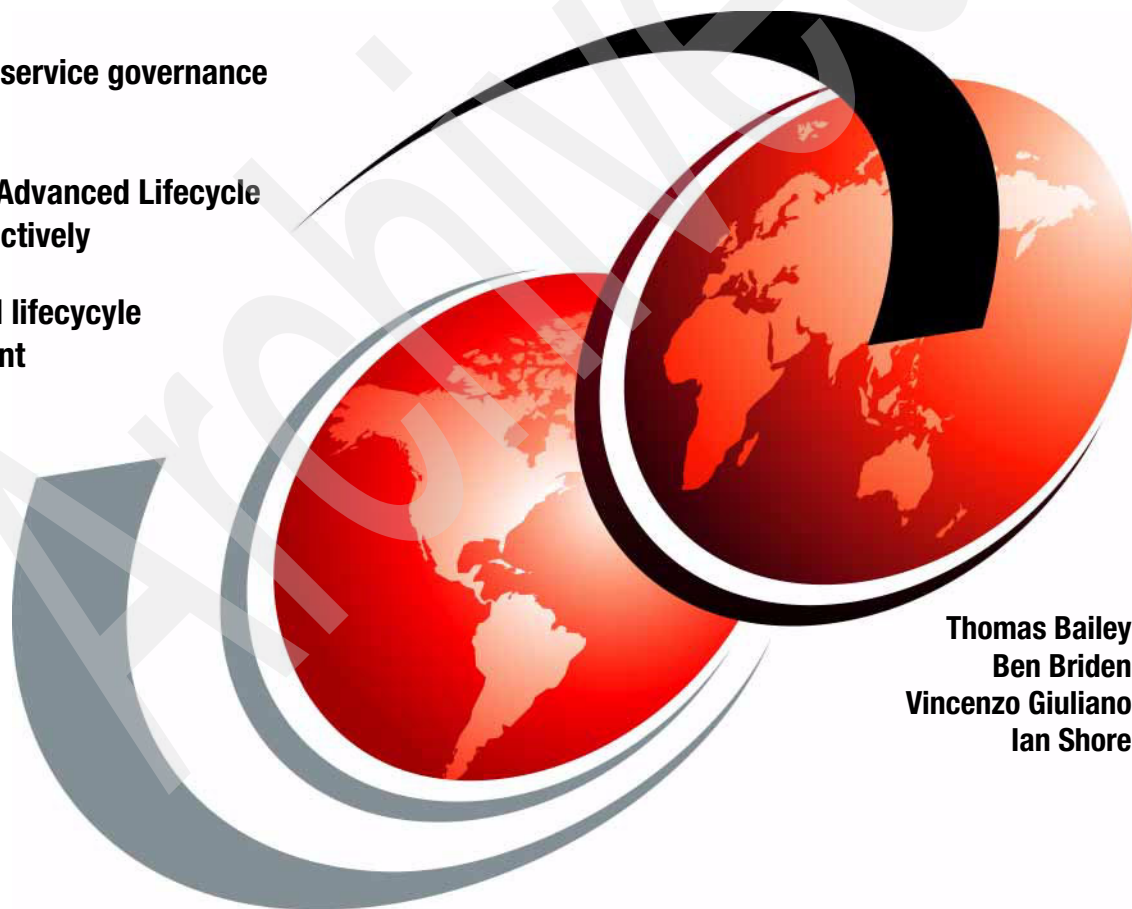


# Service Lifecycle Governance with IBM WebSphere Service Registry and Repository Advanced Lifecycle Edition

Implement service governance

Use WSRR Advanced Lifecycle  
Edition effectively

Understand lifecycle  
management



Thomas Bailey  
Ben Briden  
Vincenzo Giuliano  
Ian Shore





International Technical Support Organization

**Service Lifecycle Governance with IBM WebSphere  
Service Registry and Repository Advanced  
Lifecycle Edition**

September 2009

Archived

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

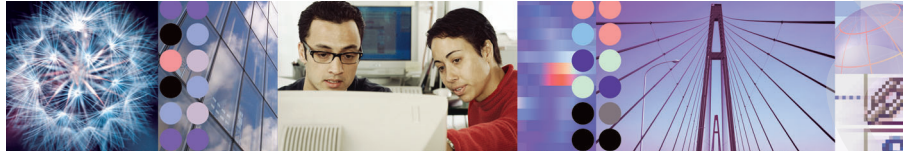
**First Edition (September 2009)**

This edition applies to WebSphere Service Registry and Repository Advanced Lifecycle Edition V6.3.

© Copyright International Business Machines Corporation 2009. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

## Contact an IBM Software Services Sales Specialist



Start SMALL, Start BIG, ... **JUST START**  
architectural knowledge, skills, research and development . . .  
**that's IBM Software Services for WebSphere.**

Our highly skilled consultants make it easy for you to design, build, test and deploy solutions, helping you build a smarter and more efficient business. **Our worldwide network of services specialists wants you to have it all!** Implementation, migration, architecture and design services: IBM Software Services has the right fit for you. We also deliver just-in-time, customized workshops and education tailored for your business needs. You have the knowledge, now reach out to the experts who can help you extend and realize the value.

For a WebSphere services solution that fits your needs, contact an IBM Software Services Sales Specialist:  
[ibm.com/developerworks/websphere/services/contacts.html](http://ibm.com/developerworks/websphere/services/contacts.html)



# Contents

<b>Contact an IBM Software Services Sales Specialist</b> .....	iii
<b>Notices</b> .....	xiii
Trademarks .....	xiv
<b>Preface</b> .....	xv
The team who wrote this book .....	xv
Become a published author .....	xviii
Comments welcome .....	xviii
<b>Part 1. SOA service governance</b> .....	1
<b>Chapter 1. Introduction</b> .....	3
1.1 Business goals .....	4
1.1.1 IBM definition of On Demand Business .....	4
1.2 Service-oriented architecture .....	5
1.2.1 Integration architecture .....	5
1.2.2 SOA business value and challenges .....	7
1.3 SOA governance .....	9
1.3.1 SOA governance context .....	10
1.3.2 Drivers of SOA governance .....	11
1.3.3 Service governance within SOA governance .....	13
1.4 Establishing SOA governance within an organization .....	14
1.4.1 SOA Center of Excellence .....	15
1.5 Lifecycles .....	17
1.5.1 SOA lifecycle .....	18
1.5.2 SOA service lifecycle .....	19
1.5.3 SOA governance lifecycle .....	20
1.6 WSRR Advanced Lifecycle Edition .....	22
1.6.1 WSRR Advanced Lifecycle Edition business value .....	22
1.6.2 Service development .....	25
1.6.3 Service deployment and management .....	27
1.6.4 Federation of capabilities .....	27
1.7 Additional resources .....	29
<b>Chapter 2. Architectural overview</b> .....	31
2.1 Introduction .....	32
2.2 WSRR overview .....	32
2.2.1 Metadata and entities .....	32

2.2.2 Assets . . . . .	34
2.2.3 Governance capabilities . . . . .	35
2.2.4 Security . . . . .	36
2.2.5 Web interface . . . . .	36
2.2.6 Integration with products . . . . .	37
2.3 Rational Asset Manager overview . . . . .	37
2.3.1 Web interface . . . . .	38
2.3.2 Assets . . . . .	39
2.3.3 Communities . . . . .	40
2.3.4 Review process . . . . .	40
2.4 WSRR Advanced Lifecycle Edition . . . . .	41
2.4.1 Synchronization . . . . .	41
2.5 Deployment topologies . . . . .	42
2.5.1 Typical WSRR installation topology . . . . .	42
2.5.2 Typical Rational Asset Manager installation topology . . . . .	44
2.5.3 Architectural deployment of WSRR Advanced Lifecycle Edition . . . . .	45
2.6 Roles and responsibilities . . . . .	46
2.6.1 GEPAAdministrator . . . . .	46
2.6.2 GEPUUser . . . . .	47
2.6.3 Business . . . . .	47
2.6.4 SOAGovernance . . . . .	47
2.6.5 Development . . . . .	47
2.6.6 Operations . . . . .	47
2.7 Service governance assets . . . . .	48
2.7.1 Entities for the Business role . . . . .	49
2.7.2 Entities for the Development role . . . . .	50
2.7.3 Entities for the SOA Governance role . . . . .	51
2.7.4 Entities for the Operations role . . . . .	52
2.8 WSRR Advanced Lifecycle Edition synchronizable assets . . . . .	52
2.8.1 Business Capability, Business Solution . . . . .	55
2.8.2 Capability Version, Release . . . . .	57
2.8.3 Service Interface Specification, Specification . . . . .	58
2.8.4 Schema Specification, Specification . . . . .	59
2.8.5 Document of Understanding . . . . .	60
2.9 Managing synchronization through the lifecycles . . . . .	60
2.9.1 Capability lifecycle . . . . .	61
2.9.2 SOA lifecycle . . . . .	63
2.9.3 Asset lifecycle . . . . .	66
2.10 Additional WSRR-defined lifecycles . . . . .	68
2.11 Additional resources . . . . .	71
<b>Chapter 3. Scenario overview . . . . .</b>	<b>73</b>
3.1 Introduction . . . . .	74



3.2	Top-down development . . . . .	74
3.3	Bottom-up development . . . . .	75
3.4	Service version management . . . . .	76
3.4.1	Compatible version management . . . . .	76
3.4.2	Incompatible version management . . . . .	77
3.4.3	Version numbering . . . . .	77
3.5	Endpoint management . . . . .	77
3.5.1	Endpoint relocation . . . . .	78
3.5.2	Retiring endpoints . . . . .	78
3.5.3	Deploying additional endpoints . . . . .	78
3.6	Reporting . . . . .	79
<b>Chapter 4.</b>	<b>JKHL Enterprises scenario overview . . . . .</b>	<b>81</b>
4.1	Introduction . . . . .	82
4.2	An overview of JKHL Enterprises . . . . .	82
4.2.1	SOA governance vision and requirements . . . . .	83
4.2.2	SOA governance solution . . . . .	84
4.2.3	Organizational structure . . . . .	86
4.2.4	Organizational structure representation in WSRR . . . . .	88
4.2.5	Organizational structure within Rational Asset Manager . . . . .	89
4.2.6	Customizations . . . . .	89
4.2.7	Governance review process . . . . .	90
4.3	Success criteria reporting . . . . .	91
4.4	JKHL Enterprises project. . . . .	91
4.4.1	JKHL Enterprises sports equipment Web site. . . . .	91
4.4.2	JKHL Enterprises sports equipment Web site upgrade . . . . .	92
<b>Part 2.</b>	<b>Top-down scenario . . . . .</b>	<b>95</b>
<b>Chapter 5.</b>	<b>Defining the business requirements . . . . .</b>	<b>99</b>
5.1	Introduction . . . . .	100
5.2	Business value definition . . . . .	100
5.2.1	Business Capability definition . . . . .	100
5.3	Progress summary . . . . .	107
<b>Chapter 6.</b>	<b>Elaboration and review of business requirements . . . . .</b>	<b>109</b>
6.1	Introduction . . . . .	110
6.2	Business application charter definition . . . . .	110
6.2.1	Business Application charter review . . . . .	114
6.2.2	Application governance review . . . . .	116
6.3	Progress summary . . . . .	119
6.4	Application scoping and investigation . . . . .	120
6.5	Creating Documents of Understanding . . . . .	125
6.6	Progress summary . . . . .	126

<b>Chapter 7. Release specification</b>	129
7.1 Introduction	130
7.2 Ordering application scope and sizing	130
7.2.1 Ordering application scope review	132
7.2.2 Ordering application development plan	135
7.2.3 Ordering application development plan review	138
7.3 Shipment service scope and sizing	141
7.3.1 Shipment service scope review	144
7.3.2 Ship Goods service development plan	146
7.3.3 Ship Goods service development plan review	150
7.4 Development plan business review	151
7.5 Ship Goods service interface and schema design	153
7.6 Ordering application interface and schema design	155
7.6.1 Ordering application interface and schema review	161
7.6.2 Ordering application interface and schema artifact development	163
7.6.3 Ordering application interface and schema review	164
7.6.4 Specification relationships	167
7.6.5 Specification asset review	171
7.7 Document of Understanding proposal	173
7.7.1 Document of Understanding scope review	174
7.7.2 Document of Understanding development review	175
7.7.3 Document of Understanding business review	177
7.8 Service Version specification	181
7.8.1 Ship Goods endpoint specification	181
7.8.2 Ordering application endpoint specification	183
7.9 Progress summary	184
 <b>Chapter 8. Defining Service Level Agreements</b>	 187
8.1 Introduction	188
8.2 Creating a Service Level Definition	188
8.2.1 Ordering application Service Level Definition	188
8.2.2 Ordering application Service Level Definition scope review	190
8.2.3 Ordering application Service Level Definition specification	191
8.2.4 Ordering application Service Level Definition review	193
8.2.5 Ship Goods Service Level Definition	194
8.2.6 Ship Goods Service Level Definition scope review	197
8.2.7 Ship Goods Service Level Definition specification	198
8.2.8 Ship Goods Service Level Definition review	200
8.3 Creating Service Level Agreements	201
8.3.1 Service Level Agreement approval	206
8.3.2 Completing the service level specifications	208
8.3.3 Service Level Agreement activation	208
8.4 Version specification approval	210

8.4.1 Ship Goods specification proposal	210
8.4.2 Ship Goods specification approval	212
8.4.3 Ordering application specification proposal	214
8.4.4 Ordering application specification approval	216
8.5 Progress summary	218
<b>Chapter 9. Release development</b>	219
9.1 Introduction	220
9.2 Development implementation	220
9.3 Ship Goods service realization	220
9.3.1 Realization approval	222
9.3.2 Staging deployment	224
9.3.3 Staging review.	231
9.3.4 Service Level Agreement activation	233
9.4 Ordering application realization.	235
9.4.1 Realization approval	237
9.5 Progress summary	239
<b>Chapter 10. Application staging</b>	241
10.1 Ordering application staging deployment	242
10.2 Ship Goods consumer verification.	248
10.3 Ordering application consumer verification	252
10.4 Application and service certification	254
10.5 Progress summary	256
<b>Chapter 11. Production deployment</b>	257
11.1 Ship Goods service production deployment	258
11.2 Ordering application production deployment.	266
11.3 Progress summary	275
<b>Part 3. Other scenarios</b>	277
<b>Chapter 12. Bottom-up scenario</b>	279
12.1 Introduction	280
12.2 Registering the Eligibility service WSDL	280
12.3 Defining the Business Capability for the Eligibility service	282
12.4 Providing retrospective scope and planning information for the Eligibility service	285
12.5 Enabling governed consumption and reuse	288
12.6 Completing the SOA service lifecycle for the Eligibility service.	290
<b>Chapter 13. Endpoint management</b>	297
13.1 Introduction	298
13.2 Problem analysis and defect resolution.	298

13.3 Staging deployment . . . . .	299
13.3.1 Staging review . . . . .	303
13.4 Production deployment . . . . .	303
<b>Chapter 14. Service Level Definition management . . . . .</b>	<b>311</b>
14.1 Introduction . . . . .	312
14.2 Scenario . . . . .	312
14.3 Consumer identification . . . . .	312
14.3.1 Consumer notification . . . . .	313
14.4 Superseding the service version . . . . .	313
14.5 Consumer response to notification . . . . .	316
14.5.1 Consumer creates new service level agreement . . . . .	316
14.5.2 Service Level Agreement approval . . . . .	319
14.5.3 Service Level Agreement activation . . . . .	321
14.5.4 Integration testing . . . . .	323
14.5.5 Production release . . . . .	323
14.6 Deprecating the Service Level Definition . . . . .	327
14.6.1 Deprecating service version . . . . .	328
14.7 Retiring a Service Level Definition . . . . .	330
14.7.1 Retiring the service version . . . . .	334
<b>Chapter 15. Reporting . . . . .</b>	<b>337</b>
15.1 Introduction . . . . .	338
15.2 WSRR report tool . . . . .	338
15.2.1 Data source . . . . .	339
15.2.2 Data set . . . . .	340
15.2.3 Joint data set . . . . .	343
15.2.4 Report table . . . . .	345
15.2.5 Report chart . . . . .	345
15.3 Business Capability reuse . . . . .	345
15.3.1 Bar chart showing Business Capability reuse . . . . .	346
15.3.2 Table showing Business Capability reuse . . . . .	353
15.3.3 Report of Business Capability reuse . . . . .	355
15.4 Business Capabilities by lifecycle state . . . . .	355
15.4.1 Pie chart for Business Capabilities by lifecycle state . . . . .	355
15.4.2 Table for Business Capabilities by lifecycle state . . . . .	357
15.4.3 Report on Business Capabilities by lifecycle state . . . . .	358
15.5 Business Capabilities by Organization . . . . .	359
15.5.1 Bar chart for Business Capabilities by Organization . . . . .	360
15.5.2 Table for Business Capabilities by Organization . . . . .	360
15.5.3 Table for Rogue Business Capabilities . . . . .	361
15.5.4 Report on Business Capabilities by Organization . . . . .	361
<b>Part 4. Appendixes . . . . .</b>	<b>363</b>

<b>Appendix A. Configuring the tooling used in the scenarios</b>	365
Software versions and topology	366
Security considerations	366
Configuration	367
WebSphere Service Registry and Repository user to role mappings	367
Rational Asset Manager	367
Rational Asset Manager review process	368
Theme customization	368
Charter validator	369
Policy customization	369
<b>Appendix B. Additional material</b>	371
Locating the Web material	371
Using the Web material	372
How to use the Web material	372
<b>Related publications</b>	375
IBM Redbooks publications	375
Online resources	376
How to get IBM Redbooks publications	376
Help from IBM	377



# 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.

The following company name that appears in this publication is fictitious:

JKHL Enterprises (JKHLE)

This name is used for instructional purposes only.

## 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

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

CICS®  
DataPower®  
DB2®  
IBM®

Rational®  
Redbooks®  
Redbooks (logo) ®  
RequisitePro®

Tivoli®  
WebSphere®  
z/OS®

The following terms are trademarks of other companies:

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

EJB, J2EE, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, SQL Server, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Eclipse and BIRT are trademarks of Eclipse Foundation, Inc.

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



# Preface

The purpose of service-oriented architecture (SOA) governance is to act as the catalyst between business governance and IT governance by ensuring optimal business value for services investments. Within SOA governance, there is service governance, which ensures that organizations build the right services, in the right way, at the right time, and then manage and reuse those services effectively. Service governance does this by overseeing the processes of proactively identifying, assessing, building, and managing high-value business services, those services that provide the greatest return on investment. Service governance requires a federated, end-to-end approach to service lifecycle governance and management, including knowing what needs to be strictly governed and where a lighter touch is more appropriate.

As SOA evolves to be a fundamental and critical enabler for other efforts, such as IBM® Smarter Planet and Cloud Computing initiatives, it becomes more imperative to ensure quality SOA and service governance capabilities. These capabilities drive discipline and value for other areas, such as the governance of provisioning of assets, impact analysis, and asset/service portfolio management.

This IBM Redbooks publication identifies the key functions and capabilities that are required for service governance based on field best practices and client scenarios. This expertise gives IBM the ability to deliver higher business value to our clients in more consumable ways and value that is fit for purpose delivery methods that match the way our clients operate.

## The team who wrote this book

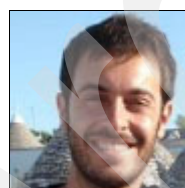
This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.



**Thomas Bailey** is the Test Architect within the WebSphere Service Registry and Repository (WSRR) development team based at the IBM Hursley Development Lab in the U.K. He has worked at IBM for eight years, the first four of which were spent working on the WebSphere Voice portfolio, and the latter four within the test team of WSRR. He has presented WSRR to many clients and explained how it fits into the IBM SOA strategy. He also works with clients as an advocate. He holds a BSc in Mathematics, Operational Research, Statistics, and Economics from the University of Warwick, and an MSc in Software Engineering from Oxford University. He is a Sun-Certified Java Programmer and holds the ISEB intermediate certificate in Software Testing.



**Ben Briden** is a Senior IT Specialist in the IBM BetaWorks organization in the U.K. He has ten years of experience with the WebSphere® product portfolio and has been working with WSRR and Rational® Asset Manager products for approximately three years. He has a bachelor degree in Computers, Management and Electronics from Salford University. After he graduated, he joined IBM. His career in IBM has ranged from software development through lab services, to these days working closely with clients, IBM Business Partners, and IBM colleagues to help them develop their knowledge and skills with the WSRR product and its various integration points.



**Vincenzo Giuliano** is in WebSphere Technical Sales at the IBM Italy S.p.A. He holds a degree in Computer Science from University degli Studi di Bari and a Masters in General Management from MIP - Politecnico di Milano School of Management. His areas of expertise include SOA, SOA governance, Java programming, and design and development of Web 2.0 applications.



**Ian Shore** is a Senior IT Specialist at the IBM Hursley Development Lab in the U.K. He has worked at IBM for 14 years and has experience with CICS® and WebSphere MQ on z/OS®, WebSphere Message Broker, WebSphere Commerce, and WSRR. During this time, he spent three years in IBM Software Services for WebSphere supporting clients with their installation and configuration of WebSphere Commerce in Europe and Africa. He is currently a development lead on the WSRR product and has presented WSRR to clients in Europe and the Middle East. He is a Sun-certified Java™ Programmer and is also familiar with Object-Oriented design, C/C++, and the Java 2 Platform, Enterprise Edition (J2EE™) platform. Ian holds a bachelor degree in Computer Science from Durham University and an M.B.A. from the University of Southampton.

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

Carla Sadtler  
International Technical Support Organization, Raleigh Center

Saurabh Agarwal  
IBM U.S.

Jorge L. Diaz  
IBM U.S.

John Falkl  
IBM U.S.

Andrew Humphreys  
IBM U.K.

Gilli Mendel  
IBM U.S.

Dennis Miller  
IBM U.S.

Laura Olson  
IBM U.S.

## Become a published author

Join us for a two-week 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](http://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](http://ibm.com/redbooks)

- ▶ Send your comments in an e-mail to:

[redbooks@us.ibm.com](mailto: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

## SOA service governance

In the first section of this part of the book, we introduce the basic concepts of service-oriented architecture (SOA), SOA governance, and SOA service lifecycle management. We then move on to explain how WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition provides the tools that are necessary to meet the needs that these architectural patterns present.

In the second section, we introduce several service lifecycle scenarios and describe the fictitious JKHL Enterprises organization on which the remaining parts of this book are built.



# Introduction

Transforming your business into an On Demand Business is a journey. Keeping down the cost of IT management and reducing IT complexity are the common challenges that you have to face in order to succeed. IBM provides the tools and expertise that you need in order to reach your goal through the use of service-oriented architecture (SOA) governance and SOA service lifecycle management.

In this chapter, we introduce the concepts of SOA, SOA governance, and SOA service lifecycle management. We discuss how you can implement these concepts by using IBM WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition:

- ▶ 1.1, "Business goals" on page 4
- ▶ 1.2, "Service-oriented architecture" on page 5
- ▶ 1.3, "SOA governance" on page 9
- ▶ 1.4, "Establishing SOA governance within an organization" on page 14
- ▶ 1.5, "Lifecycles" on page 17
- ▶ 1.6, "WSRR Advanced Lifecycle Edition" on page 22

## 1.1 Business goals

Regardless of your industrial sector, your business typically faces two questions:

- ▶ How can I increase revenue?
- ▶ How can I reduce costs?

From a tactical perspective, it seems that these questions conflict with each other: on one side, you need to invest money, and on the other side, you need to constrain budgets. For a short period, a reasonable solution can be to find a trade-off by focusing on shifting the money in the right way. But what about thinking strategically? From that perspective, the questions are related to new business opportunities and incoming threats, so you must be able to design a business that is responsive and agile in real time. No doubt about it, the solution is called On Demand Business.

### 1.1.1 IBM definition of On Demand Business

An *On Demand Business* is an enterprise whose business processes - integrated end-to-end across the company and with key partners, suppliers, and clients - can respond with speed to any client demand, market opportunity, or external threats.

At its core, an On Demand Business has these characteristics:

- ▶ Leveraging technology to enable organizations to achieve flexibility and adaptability to client demand, market opportunity, and external threats in real time
- ▶ Connecting people, processes, and information with ease
- ▶ Streamlining and integrating processes seamlessly and securely; not only within the business organization, but among business partners, vendors, and clients
- ▶ Simplifying the infrastructure with efficient monitoring, management, provisioning, and virtualization capabilities
- ▶ Increasing throughput by utilizing existing assets, capabilities, and infrastructure to the maximum
- ▶ Leveraging industry standard-based computing models to achieve all of these objectives

Becoming an On Demand Business is an evolutionary initiative that happens over a period of time. The success lies in your ability to incrementally build an On Demand operating environment that is tightly aligned with your organization's



business goals and processes to help transform your business to become an On Demand Business.

The potential benefits of the On Demand Business model are great, and the path to get you there does not need to be complex or risky. It is a matter of starting and continuing with incremental projects based on the needs of your business.

In the On Demand Business, the choice of technology is only part of the equation. At its heart is the point at which technology and business models intersect to provide new capabilities and client experiences. Remember that the extent of the technology investment for the On Demand Business must be balanced with regard to the Return On Investment (ROI) and the Total Cost of Ownership (TCO).

SOA can help you to get real business value from IT. SOA is defining an architectural approach that assists in the flexible integration of IT systems supporting your business. In the following sections, we provide an overview of several elements in SOA that can help you achieve this goal.

For more information, refer to *The Solution Designer's Guide to IBM On Demand Business Solutions*, SG24-6248.

## 1.2 Service-oriented architecture

SOA plays a crucial role for companies trying to implement the IBM vision of On Demand Business. The IBM On Demand Business vision is designed to enable clients to succeed in an environment with an unprecedented rate of change. In an on demand world, companies need to be able to respond to any client requirement, opportunity, or threat quickly and easily. To succeed in this environment, a company must be able to implement new processes quickly while leveraging existing investments. From a business perspective, On Demand Business is about providing a way for companies to realign their business and technology environment to match the need for reusable business functionality.

SOA can be seen as an architectural enabler for On Demand Business.

### 1.2.1 Integration architecture

SOA includes an integration architecture approach that is based on the concept of a *service*. The business and infrastructure functions that are required to build distributed systems are provided as services that collectively, or individually, deliver application functionality to either user applications or other services. SOA specifies that within any given architecture, there must be a consistent

mechanism for services to communicate. That mechanism needs to be loosely coupled and support the use of explicit interfaces. SOA brings the benefits of loose coupling and encapsulation to integration at an enterprise level.

Services are the building blocks to SOA, providing function out of which distributed systems can be built. Services can be invoked independently by either external or internal service consumers to process simple functions, or they can be chained together to form more complex functionality and quickly devise new capabilities. By adopting an SOA approach, companies can build flexible systems that implement changing business processes quickly, and make extensive use of reusable components.

Figure 1-1 illustrates a company that wants to implement a new business process to support clients that place orders from a Web site. The company already has existing retail, warehouse, and billing systems. It wants to build the new process by reusing the functionality that is provided by those systems rather than by writing new applications or by building new interfaces to the existing systems. If the company has already adopted an SOA approach, it will have defined the interfaces to its existing systems in terms of the functions, or services, that they can offer to support building business processes. This approach facilitates integration with their partners and vendors. The only remaining requirement is an application that makes calls to the services to complete the new business process.

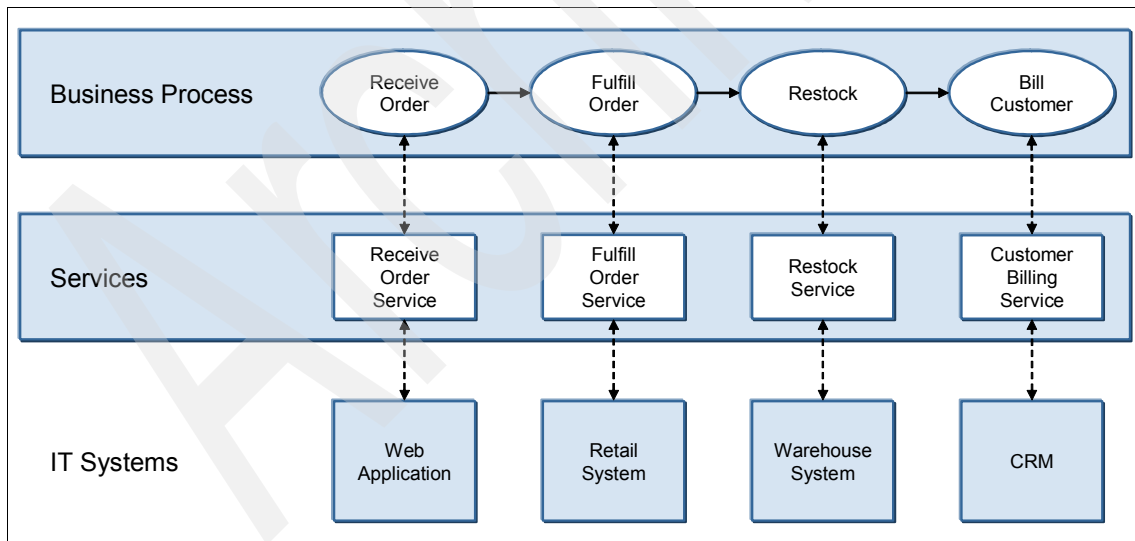


Figure 1-1 Service-oriented approach to building systems

You can implement an SOA using an approach that is tailored to your business needs. A *top-down* approach starts by identifying the business processes and business services that are used by business users. A *bottom-up* approach focuses on the reuse of your applications as services. A *meet-in-the-middle* approach is a combination of the two previous approaches. These approaches allow companies to build horizontal business processes, integrating systems, people, and processes from across the enterprise quickly and easily in response to changing business needs.

Figure 1-1 on page 6 shows that existing systems can be used to implement new business processes that extend the use of the system beyond the processes that they were originally written to support. Therefore, the company is able to maximize its previous IT investment by reusing existing IT systems without having to invest extensively to build new interfaces to the systems.

## 1.2.2 SOA business value and challenges

SOA allows you to define an architectural approach that assists in the flexible integration of IT systems to support business. The business value of SOA includes these benefits:

- ▶ Increasing the speed at which businesses can implement new products and processes, can change existing ones, or can recombine them in new ways
- ▶ Reducing the implementation and ownership costs (TCO) of IT systems and the integration between them
- ▶ Enabling flexible pricing models by outsourcing more fine-grained elements of the business than were previously possible or by moving from fixed to variable pricing based on transaction volumes
- ▶ Simplifying the integration work that is required by mergers and acquisitions
- ▶ Increasing IT utilization and ROI

Figure 1-2 on page 8 depicts the SOA Foundation Reference Architecture solution view that is used to decompose an SOA design. For example, with the top-down approach, services and business processes are decomposed into service components. *Service components* include a detailed set of definition metadata that is used to describe the service to the information system.

Services can be aggregated with various levels of granularity. Part of this aggregation process is to establish related design concerns and to determine which teams will collaborate to implement the related services to be deployed as a single unit. The resulting set of atomic and composite services will be the building blocks of your business processes. Then, that logical architecture is mapped to a physical architecture.

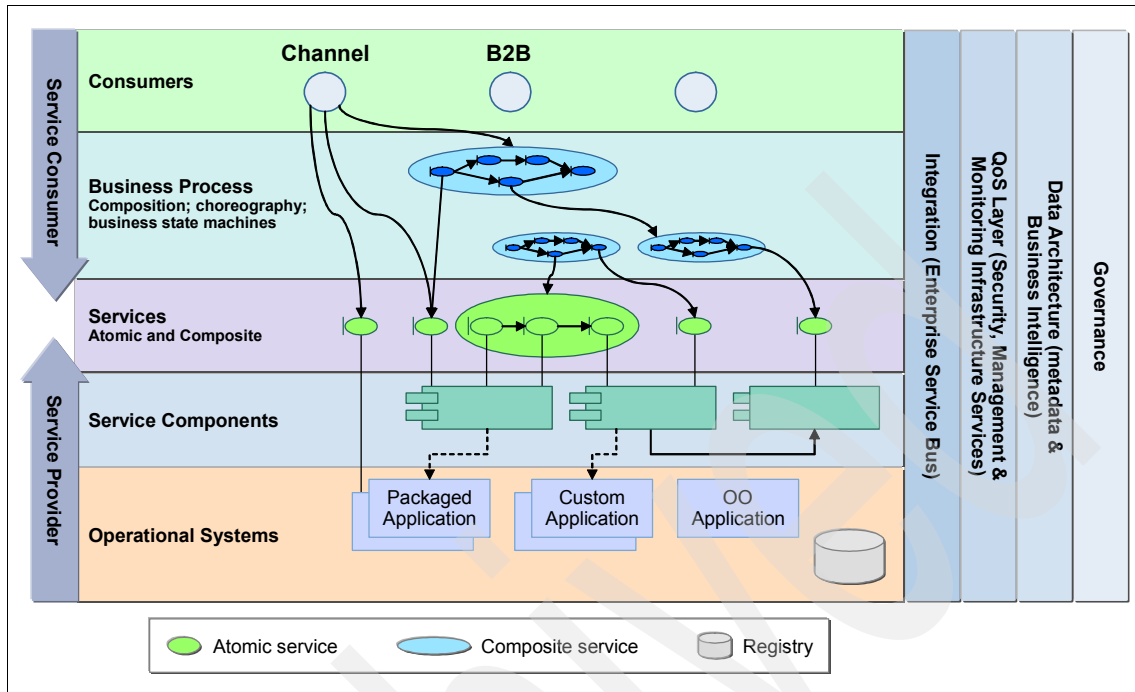


Figure 1-2 SOA solution view

In addition to offering business and IT benefits, SOA also raises a number of questions:

- ▶ Who is in charge of service development?
- ▶ Who defines the priorities of services that need to be developed?
- ▶ Is the quality of service (QoS) always ensured?
- ▶ Are services compliant with the enterprise policies?

These types of concerns must be addressed in order to control the complexity that SOA can bring into an IT environment. SOA governance provides the ability to address these issues, ensuring you continue to reap the business value of using SOA.

For more information, refer to *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240.

## 1.3 SOA governance

In practice, SOA governance guides the development and management of reusable services, establishing how services will be designed and developed and how those services will change over time. It establishes guidelines for the agreements made between the providers of services and the consumers of those services that tell the consumers what they can expect and the providers what they are obligated to provide.

SOA governance does not design the services, but it guides how the services will be designed. It helps to answer many difficult questions that are related to SOA:

- ▶ What services are available?
- ▶ Who can use them?
- ▶ How reliable are they?
- ▶ How long will they be supported?
- ▶ Can you depend on them to not change?
- ▶ What if you want them to change, for example, to fix a code problem? Or, to add a new feature?
- ▶ What if two consumers want the same service to work differently?
- ▶ Just because you decide to expose a service, does that mean you are obligated to support it forever?
- ▶ If you decide to consume a service, can you be confident that it will not be shut down tomorrow?

SOA governance builds on existing IT governance techniques and practices. Everyone thinks reusable assets are good, but they are difficult to make work in practice:

- ▶ Who is going to pay to develop them?
- ▶ Will development teams actually strive to reuse them?
- ▶ Can everyone really agree on a single set of behaviors for a reusable asset, or will everyone have their own customized version, which is not really reuse after all?

SOA and services make these governance issues even more important and thus, their consequences even more significant. Governance does not determine what the results of decisions are, but what decisions must be made and who will make them.

The two parties, the consumers and the providers, have to agree on how they will work together. Much of this understanding can be captured in a service level

agreement (SLA), including measurable goals that a service provider agrees to meet and that a service consumer agrees to live with. This agreement is like a contract between the parties, and it can be a legal contract. At the very least, the SLA articulates what the provider must do and what the consumer can expect.

### 1.3.1 SOA governance context

SOA governance has relationships to other forms of governance. There are two forms of governance that are relevant to the definition of SOA governance:

- ▶ Business Governance:
  - Establishing chains of responsibility, authority, and communication to empower people (decision rights)
  - Establishing measurement, policy, and control mechanisms to enable people to carry out their roles and responsibilities
- ▶ IT governance:
  - Establishing decision making rights associated with IT
  - Establishing mechanisms and policies that are used to measure and control the way that IT decisions are made and carried out.

SOA governance is the intersection of Business and IT governance focused on the lifecycle of services to ensure that the business value of SOA is achieved. It is the effective management of this lifecycle that is the key goal to SOA governance.

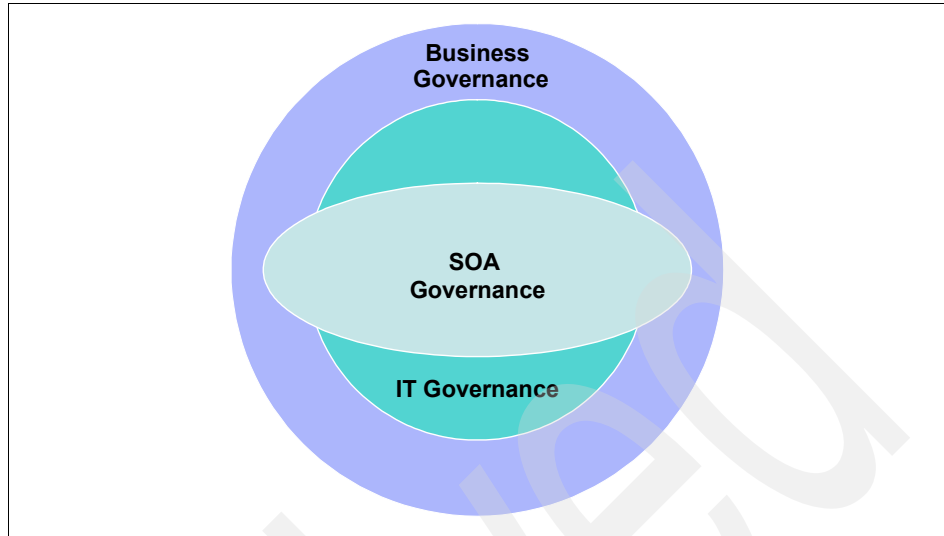


Figure 1-3 SOA governance context

### 1.3.2 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? We discuss several key drivers:

- 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 lifecycle

costs. Governance allows us to define these expectations through 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 (LoB) 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.

Referring to Figure 1-2 on page 8, to get business value from SOA and to reach the separation of concern among business needs and technical details, the companies need to define a service portfolio (atomic and composite service layer). In other words, provide the company with the following capabilities:

- ▶ 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 the IT operations team understand, discover, and maintain the relationships between services and the IT resources that support them?

- ▶ Manage services as resources

To achieve the 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 SLA. Who in the organization is responsible for this management and how do they fulfill their responsibilities?

- ▶ Identify the services to manage

A key challenge of service governance 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?

These capabilities explain why there is the need of Service Governance.

For more information, refer to *Implementing Technology to Support SOA Governance and Management*, SG24-7538, and *WebSphere Service Registry and Repository Handbook*, SG24-7386.



### 1.3.3 Service governance within SOA governance

Service governance is governing the process of service lifecycle management to maximize how that particular service delivers business value and fulfills the goals of the business. The following descriptions provide a brief overview of the various aspects of service governance.

#### **Service definition**

Regardless of whether the service is atomic or composite, it exists to satisfy a specific business need. That means its functionalities, behavior, and interface must have been agreed to across company LoBs. One of the most important aspects of service governance is overseeing the creation of the service.

#### **Service lifecycle**

Business flexibility is achieved when a company can change its services when business needs change. This change is impossible without control and without a well-defined service lifecycle to plan, design, implement, deploy, maintain, and decommission the services.

#### **Service versioning**

Usually, as soon as a service is made available, the users of that service start needing changes. Therefore, companies need the ability to provide the right service version to the right consumer.

#### **Service migration**

Using service versioning creates the need for a management approach to plan periodic migration to newer service versions. And, it is a catalyst to get more runtime reliability where the consumer request does not depend on a specific version of service.

#### **Service registry**

During both development time and deployment and management time, the role of a service registry is fundamental. We introduce this topic in 1.6, “WSRR Advanced Lifecycle Edition” on page 22.

#### **Service message model**

Business and IT alignment is reached when data types and messages mirror the business information that is defined among the LoBs. A common approach to achieve this mirroring and to avoid the chaos encouraged by not having it is to use a canonical data model.

### **Service monitoring**

When the company defines a service, it also defines an SLA among the stakeholders. Service providers must be monitored to ensure that they are meeting their associated SLAs.

### **Service ownership**

Who is responsible for each of the services? Where does this information reside? How can the company validate that a service is updated by an appropriate person? By establishing relevant ownerships to specific services, a company can control who does what to what and when.

### **Service testing**

Is your company able to test the services in an appropriate environment? A proxy that intercepts messages can reroute specific inbound messages to a test service as opposed to an existing deployed version of the service (*service virtualization*). In this way, the service in test can be tested in an environment similar to where it will be deployed.

### **Service security**

Appropriate service security ensures that an authenticated consumer is authorized to access and invoke the service that it is attempting to consume. Providing the right service to the right consumer is key to service security.

## **1.4 Establishing SOA governance within an organization**

The implementation of an SOA, and therefore of SOA governance, creates a significant amount of organizational change - including shifts in the business model, refinements to the operating model, and elimination of many functional or operational silos. The archetype that defines how the organization is run typically changes in order to reap the benefits of the SOA.

Elements of organizational change include governance planning, talent management, service ownership, business responsiveness, and organization redesign. All of these elements must occur if the organization is to achieve agility from its SOA. These changes must be identified and planned in the organizational change management approach so that they can be implemented in the governance model (Table 1-1 on page 15).

Table 1-1 SOA governance model

<b>Principles</b>	Guiding objectives' and goals' associated metrics to ensure that they are met.
<b>Role and responsibilities</b>	The roles and associated responsibilities that will facilitate business and IT alignment and properly establish decision rights.
<b>Guiding decisions</b>	Policies, guidelines, best practices, and standards. Documenting these decisions can be through human or machine (for automation) language.
<b>Methods</b>	Consistent approach to establishing SOA governance and applying it across the lifecycle.
<b>Foundational governance processes</b>	Exceptions and appeals. Compliance. Communication.
<b>Platform</b>	Enabling technology (for example, registry and repository, monitoring, and so forth).

In this section, we identify a candidate set of teams required to establish or to refine an organizational SOA governance and management capability. In addition to describing the teams, we outline any specific tasks for SOA governance.

### 1.4.1 SOA Center of Excellence

A recommended practice for realizing a SOA governance effort within an enterprise is through a Center of Excellence (CoE). A CoE is a board of knowledgeable SOA practitioners who establish and supervise policies to help ensure that an enterprise is successful with SOA (Figure 1-4 on page 16). The CoE establishes policies for the identification and development of services, establishment of SLAs, management of registries, and other efforts that provide effective governance. CoE members then help put those policies into practice, mentoring and assisting teams with developing services and composite applications.

To be effective, the CoE must have the authority to put these policies in place and to enforce them. This authority often means sponsorship from outside the IT department and control over the budgets required to put the policies in place.

After the governance CoE works out the policies, technology can be used to manage those policies. Technology does not define an SLA, but it can be used to enforce and measure compliance. For example, technology can limit which

consumers can invoke a service and when they can do so. It can warn a consumer that the service has been deprecated. It can measure the service's availability and response time.

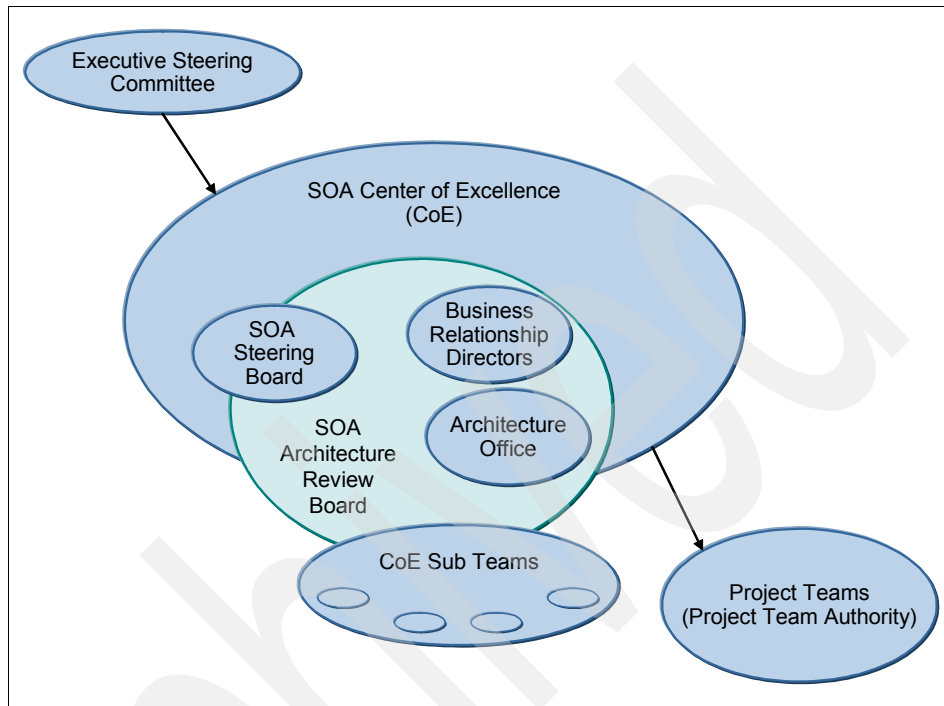


Figure 1-4 A typical SOA CoE structure

The following list provides a brief description of each related responsibility:

- ▶ Executive Steering Committee:
  - Sets business direction and priorities
  - Authorizes funding
  - Enforces SOA mandate
  - Resolves disputes (final)
- ▶ Center of Excellence:
  - Leads and manages the IT investments
  - Takes decisions about the design and the implementation of strategic SOA solutions
- ▶ SOA Steering Board:
  - Sets SOA vision, principles, and policies
  - Prioritizes SOA efforts
  - Allocates SOA funding

- Resolves exception requests
- ▶ Business Relationship Directors:
  - Communicate business needs and priorities
  - Key touch point with LoBs, such as business unit leader or business analyst
- ▶ Architecture Office:
  - Sets SOA standards and policies
  - Defines all SOA architecture elements: technologies and applications
  - Provides thought leadership
  - Harvests assets
  - Supports SOA development and operations
  - Permanent roles
- ▶ SOA Architecture Review Board:
  - Decides the technology standards/best practices to meet business needs
  - Decides when a change or addition in the SOA is needed
  - Prioritizes initiatives
- ▶ CoE Sub Teams:
  - Pool of skilled resources
  - Execute day-to-day SOA functions
  - Rotational roles
- ▶ Project Teams:
  - Develop applications using SOA assets under the guidance of the CoE
  - Ensure project-level compliance

## 1.5 Lifecycles

The On Demand Business paradigm is greatly realized through SOA and its control. For this reason, you need a well-defined service lifecycle, which helps you to manage (and will allow for the construction of) service-based ecosystems that enable efficient transformations of business environments.

In the next sections, we provide a brief overview of SOA, SOA governance, and SOA service lifecycles.

## 1.5.1 SOA lifecycle

IBM clients have indicated that they think of SOA in terms of a lifecycle. As you can see in Figure 1-5, the IBM SOA Foundation includes the following lifecycle phases:

- ▶ Model
- ▶ Assemble
- ▶ Deploy
- ▶ Manage

There are two key points to consider about the SOA lifecycle: First, the SOA lifecycle phases apply to all SOA projects. Second, the activities in any part of the SOA lifecycle can vary in scale and the level of tooling that is used, depending on the stage of adoption.

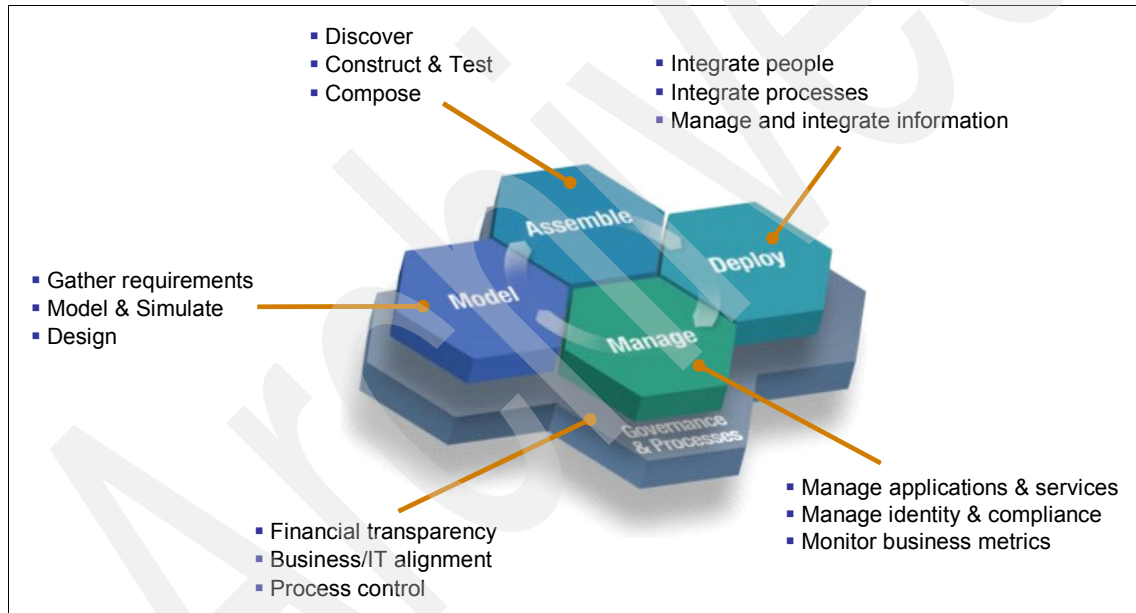


Figure 1-5 SOA lifecycle

### Model

*Modeling* is the process of capturing the business design from an understanding of the business requirements and objectives. The business requirements are translated into a specification of business processes, goals, and assumptions for creating a model of the business.

## Assemble

During this step, you can use your business design to communicate with the IT organization to assemble the information system artefacts that will implement the business design.

## Deploy

The deploy phase of the lifecycle includes a combination of creating the hosting environment for the applications and the deployment tasks of those applications. This phase includes resolving the application's resource dependencies, operational conditions, capacity requirements, and integrity and access constraints.

## Manage

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

### 1.5.2 SOA service lifecycle

The SOA lifecycle spans the service lifecycle. The goal is to maximize the business value and prioritize the coordination of each and every service. In Table 1-2 on page 20 and in Figure 1-6, you can see a brief example of a service lifecycle.

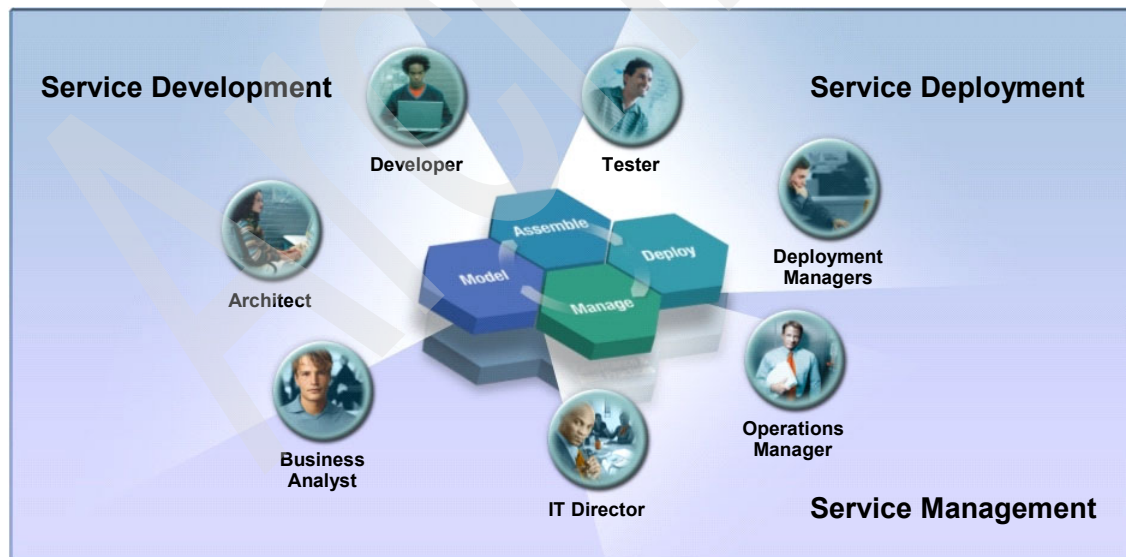


Figure 1-6 SOA service lifecycle

Table 1-2 A brief description of SOA service lifecycle

<b>Service Development (Model, Assemble)</b>	<b>Service Deployment (Deploy)</b>	<b>Service Management (Manage)</b>
<ul style="list-style-type: none"> <li>▶ Identify the service business domain</li> <li>▶ Model the service: <ul style="list-style-type: none"> <li>– Identify the service</li> <li>– Specify the service</li> <li>– Realize the service</li> </ul> </li> <li>▶ Design the service architecture</li> <li>▶ Develop the service</li> <li>▶ Test service functionalities</li> </ul>	<ul style="list-style-type: none"> <li>▶ Deploy the service</li> <li>▶ Test the service within the infrastructure</li> </ul>	<ul style="list-style-type: none"> <li>▶ Monitor the service performance</li> <li>▶ Monitor the access to the service</li> <li>▶ Manage the service changes</li> <li>▶ Manage the retirement of the service</li> </ul>
<b>Governance (Plan, Define)</b>	<b>Governance (Enable)</b>	<b>Governance (Measure)</b>
<ul style="list-style-type: none"> <li>▶ Define service focus</li> <li>▶ Identify service owners</li> <li>▶ Define and refine policies, standards, and QoS</li> <li>▶ Make service realization decisions</li> </ul>	<ul style="list-style-type: none"> <li>▶ Initiate and enable the notification and validation mechanisms that are related to service deployment</li> <li>▶ Enforce policies</li> </ul>	<ul style="list-style-type: none"> <li>▶ Measure the QoS</li> <li>▶ Measure the service security</li> <li>▶ Measure the service conformity with policies</li> </ul>

### 1.5.3 SOA governance lifecycle

SOA governance must optimize the value of IT in achieving business outcomes. The IBM SOA Governance and Management Method (SGMM) defines the SOA governance and management lifecycle as shown in Figure 1-7 on page 21. This lifecycle consists of four phases:

- ▶ Plan
- ▶ Define
- ▶ Enable
- ▶ Measure



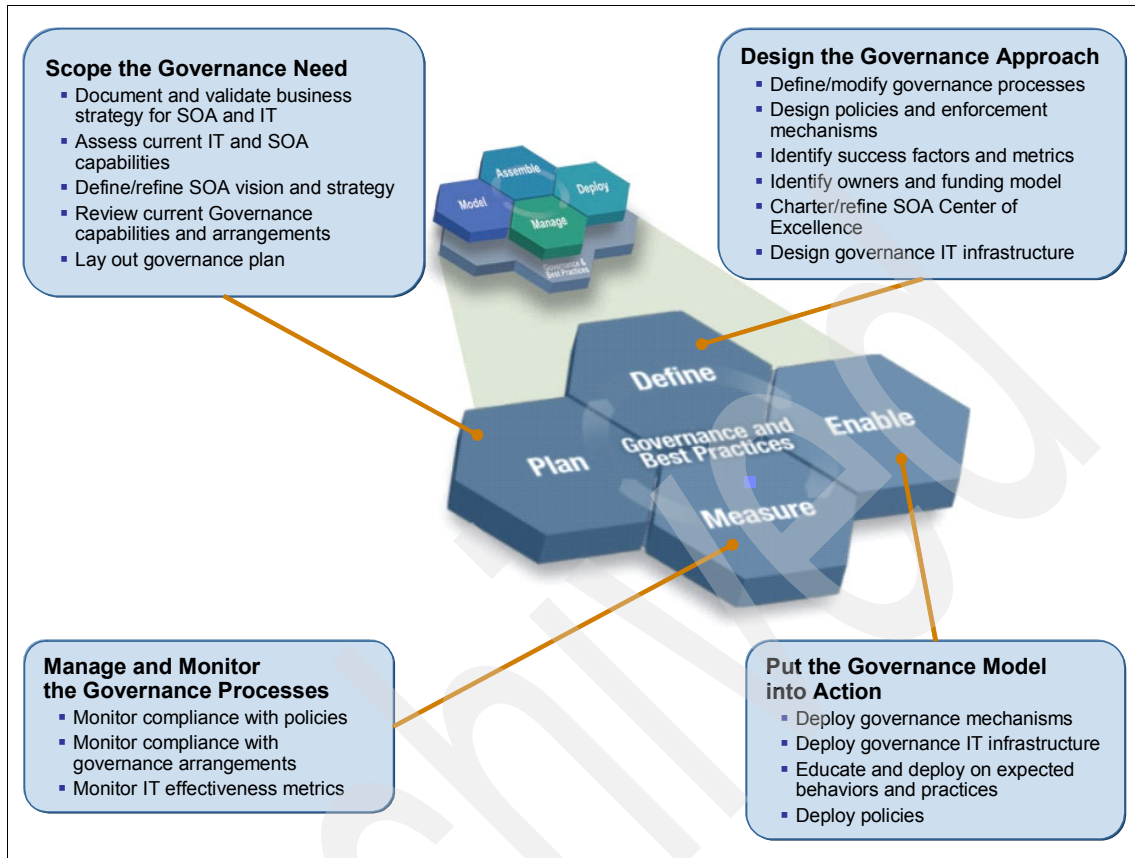


Figure 1-7 SOA governance lifecycle

## Plan

During the planning phase of building an SOA governance framework, you focus on understanding the overall scope of the governance need within your organization and on identifying areas for improvement.

## Define

When the opportunities for improved governance are identified, business and IT professionals work together to define and modify the current governance arrangements and mechanisms. For example, agree to new approaches to creating policies at this time.

## Enable

Solutions to governance needs are put into action during this phase of establishing the SOA governance framework. These solutions can include the

deployment of new or enhanced governance arrangements. Governance activities within this phase influence how SOA is deployed by enabling the policy enforcement infrastructure.

### **Measure**

During this phase, governance arrangements and mechanisms that were identified in the Define phase and deployed to the Enable phase are monitored.

## **1.6 WSRR Advanced Lifecycle Edition**

WebSphere Service Registry and Repository Advanced Lifecycle Edition is a powerful enabler of SOA and Service Governance. It provides a number of mechanisms and features to implement the vision of a fully controlled SOA environment.

### **1.6.1 WSRR Advanced Lifecycle Edition business value**

As the integration point for service metadata, WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition establishes a central point for finding and managing service metadata and assets acquired from a number of sources, including service application deployments and other service metadata and endpoint registries and repositories, such as UDDI.

WSRR Advanced Lifecycle Edition provides combined, integrated design time and runtime SOA repositories to govern the service lifecycle from creation to consumption. This capability is powered by these products:

- ▶ IBM Rational Asset Manager, which manages information that is useful for developing, reusing, and managing all types of reusable assets
- ▶ IBM WebSphere Service Registry and Repository, which manages information that is useful for the runtime operation, management, and deployment use of services

WSRR Advanced Lifecycle Edition provides these features:

- ▶ A scalable, flexible enterprise-level solution that can manage services from inception to deployment
- ▶ Integrated repositories to store, catalog, and flexibly organize assets and services
- ▶ Customizable workflow for approvals and service governance lifecycle
- ▶ Flexible solution for capturing relationships between services and support for impact analysis

- ▶ Ability to report on key metrics to support planning and assessment
- ▶ Solution that integrates with your SOA design and runtime systems, including other existing repositories

## Features and benefits

Table 1-3 briefly lists the features and benefits that you have for SOA service lifecycle management by using WSRR Advanced Lifecycle Edition.

*Table 1-3 WSRR Advanced Lifecycle Edition features and benefits*

Benefits	Service Development	Service Deployment
Store and Organize	<ul style="list-style-type: none"> <li>▶ Easily locate existing reusable assets</li> <li>▶ Create and package assets</li> <li>▶ Fast search using categories, facets, or keywords</li> <li>▶ Custom categorization for assets</li> </ul>	<ul style="list-style-type: none"> <li>▶ Publish and find your services and related information in one place</li> <li>▶ Increase visibility of services throughout your organization</li> <li>▶ Expose redundant and inefficient services</li> </ul>
Customize Workflow	<ul style="list-style-type: none"> <li>▶ Easily approve, review, rate, and discuss assets</li> <li>▶ Define workflow associated with creating and reviewing assets</li> <li>▶ Implement roles and access controls for assets at development time</li> </ul>	<ul style="list-style-type: none"> <li>▶ Implement decision rights and processes to make changes</li> <li>▶ Easily approve, promote, introduce, and retire services</li> <li>▶ Maintain test environment separate from production</li> </ul>
Impact Assessment	<ul style="list-style-type: none"> <li>▶ Ratings, feedback, and statistics help in asset evaluation</li> <li>▶ RSS Feeds, e-mail, and discussion groups notify users of changes to assets</li> <li>▶ Traceability to related assets and service implementation artifacts</li> </ul>	<ul style="list-style-type: none"> <li>▶ Expose multiple versions of the same service</li> <li>▶ Efficiently manage changes to service metadata and notify stakeholders</li> <li>▶ Graphical impact analysis to understand service interdependencies</li> </ul>

Benefits	Service Development	Service Deployment
Audit and Report	<ul style="list-style-type: none"><li>▶ Comprehensive asset-level reporting and metrics</li><li>▶ Asset activity and audit reporting</li><li>▶ Report on an asset at the repository level</li></ul>	<ul style="list-style-type: none"><li>▶ Plan and assess the service lifecycle workflow</li><li>▶ Plan and assess the services and business processes managed through the service lifecycle</li></ul>

Furthermore, you need to manage information that is useful for managing and tracking all types of deployed software components, such as the topology and environment in which service endpoints exist. Typically, this information is stored in the Change and Configuration Management Database (CCMDB). To fully support the SOA service lifecycle management, WSRR Advanced Lifecycle Edition provides you the ability to federate with the CCMDb to allow full control over your services.

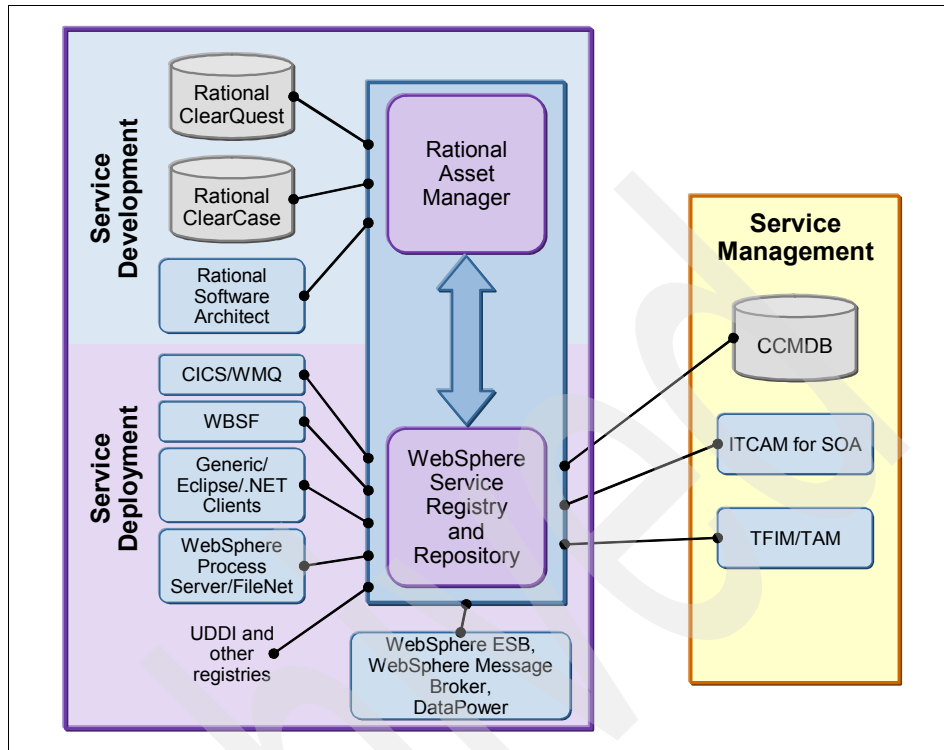


Figure 1-8 Integral to SOA

## 1.6.2 Service development

At the most basic level during development, an SOA consists a service provider and service consumer. As soon as the number of the services in use starts to grow (for example, more than 20 services), a service registry is usually added to the SOA (Figure 1-9 on page 26).

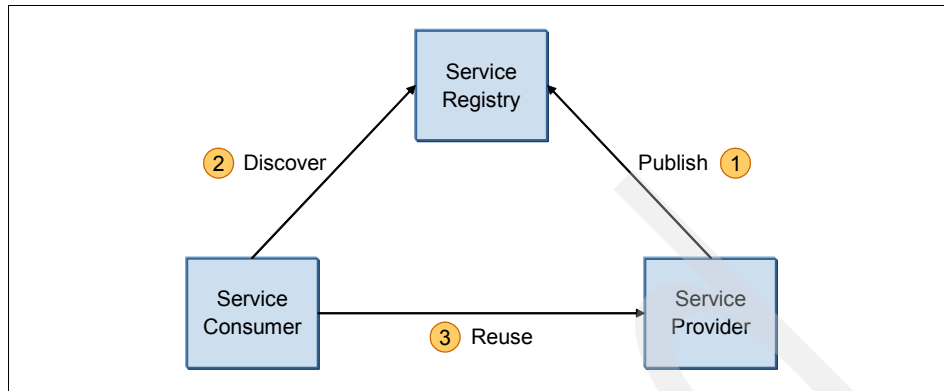


Figure 1-9 SOA components at the design time

The *service provider* creates a service and, in certain cases, publishes its interface and access information to a service registry. Each provider must decide which services to expose, evaluate trade-offs between security and easy availability, determine how to price the services, or determine how to exploit the value of the services if they are free. The provider also has to decide in which category to list the service and what sort of trading partner agreements are required to use the service.

The *service registry* is responsible for making the service interface and implementation access information available to service consumers. The implementers of a service registry must consider the scope with which the registry will be implemented. For example, there are public service registries available over the Internet to an unrestricted audience, as well as private service registries that are only accessible to users within a company-wide intranet.

The *service consumer* locates (discovers) entries in the service registry and then binds to the service provider in order to invoke the defined service.

Even with this simple scenario, there are several interesting questions:

- ▶ How can we notify service consumers of the presence of new services? And in general, how can we improve the reuse of services?
- ▶ How can we eliminate the junk services? Are we able to limit publishing of services to only those services that are necessary for the business?
- ▶ Is the service provider compliant with the enterprise policy?

A simple registry that can publish and search services cannot address these questions.

A service consumer can also act as a service provider and the reverse is also true. For instance, if a service provider needs additional information that it can only acquire from another service, it acts as a service consumer.

### 1.6.3 Service deployment and management

At the most basic level, during the deployment and management, an SOA consists of the same components that are presented at the design time, along with an additional component, which is the Broker (Figure 1-10).

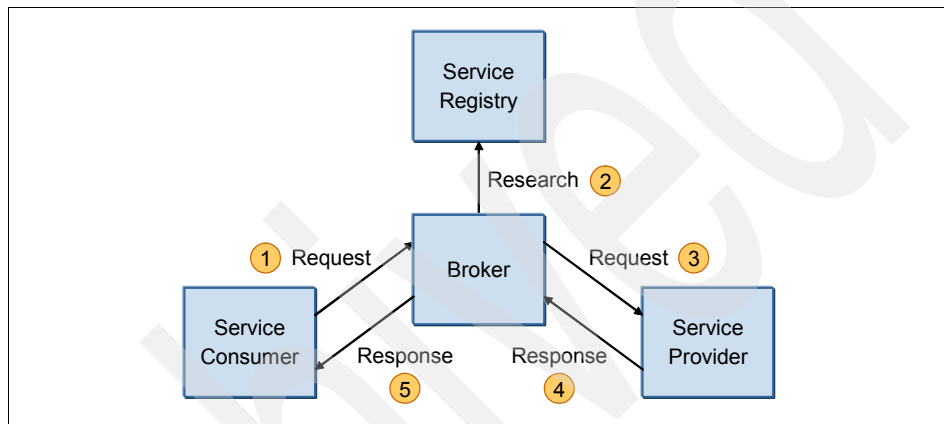


Figure 1-10 Basic SOA component at run time

The job of the *broker* is to accept the consumer request, to look for the service provider, to route the consumer request to the selected provider, eventually to transform the message formats, and to deliver the provided response.

Again, there are issues to consider:

- Is the broker able to manage the service when it is unavailable?
- How can the broker choose the right provider? Is the broker able to understand which provider is performing well?
- Is the runtime environment always compliant with the enterprise policies?

A broker working alone, or a broker working along with a simple registry, cannot easily address these important issues.

### 1.6.4 Federation of capabilities

As you have seen in the previous sections, the SOA, the SOA governance, and service governance are pervasive in your company's organization. We have

shown that a simple registry alone is insufficient. Even a single repository is not enough.

You need a federation of capabilities (Figure 1-11) to support your service governance.

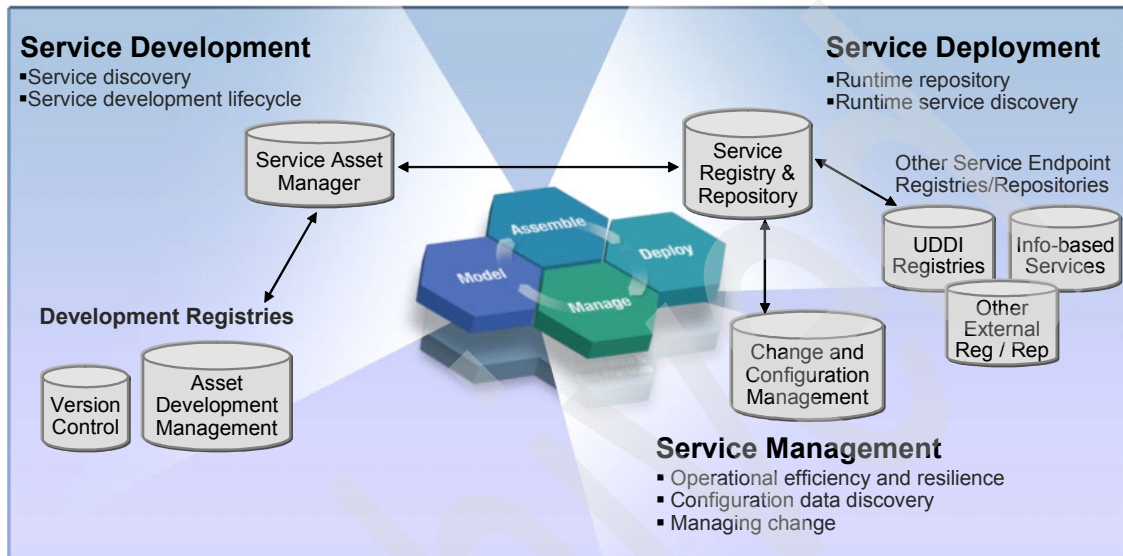


Figure 1-11 Federation of capabilities

Indeed, a business needs multiple service and asset registries or repositories that will need to be federated. Refer to Table 1-2 on page 20 where we define briefly the federated capabilities that you need for end-to-end SOA service lifecycle management.

Table 1-4 Federation of capabilities

Service Development	Service Deployment	Service Management
<ul style="list-style-type: none"> <li>Define asset types</li> <li>Create and manage all types of assets</li> <li>Provide asset traceability and details</li> <li>Collaborate on asset development</li> </ul>	<ul style="list-style-type: none"> <li>Select a service endpoint dynamically at run time</li> <li>Govern runtime changes to service metadata</li> <li>Set runtime policies for service execution</li> <li>Receive deployed service details</li> </ul>	<ul style="list-style-type: none"> <li>Capture both actual and authorized application and service data and topologies</li> <li>Establish relationships and references between all logical/physical entities</li> <li>Record configuration relationships using metadata</li> </ul>



A registry, for example, Universal Description, Discovery, and Integration (UDDI), can be useful if you need to publish and find services. But if you need to establish a central point for finding and managing service metadata acquired from a number of sources, if you need to control the service visibility, to manage versions, to analyze and communicate proposed changes, and to monitor usage of the SOA foundation, you need more than a service registry.

You really need a registry and a repository able to manage the SOA service lifecycle.

## 1.7 Additional resources

For more information about these topics, see these resources:

- ▶ Introduction to SOA governance  
<http://www.ibm.com/developerworks/library/ar-servgov/>
- ▶ SOA Governance and Service Lifecycle Management  
<http://www.ibm.com/soa/gov>
- ▶ WebSphere Service Registry and Repository Advanced Lifecycle Edition  
[http://www-01.ibm.com/software/integration/wsrr/ale/about/?S\\_CMP=rnav](http://www-01.ibm.com/software/integration/wsrr/ale/about/?S_CMP=rnav)
- ▶ *The Solution Designer's Guide to IBM On Demand Business Solutions*, SG24-6248
- ▶ *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240
- ▶ *Implementing Technology to Support SOA Governance and Management*, SG24-7538
- ▶ *WebSphere Service Registry and Repository Handbook*, SG24-7386

Archived

## Architectural overview

In this chapter, we provide an architectural overview of deploying WebSphere Service Registry and Repository (WSRR) with Rational Asset Manager and the manner in which they form WSRR Advanced Lifecycle Edition.

We discuss the expected roles that are available in WSRR Advanced Lifecycle Edition along with their responsibilities. We also discuss the relevant assets that can be synchronized, the relationships among those assets, and how and when the assets are synchronized.

We address these topics in the following sections within this chapter:

- ▶ 2.1, “Introduction” on page 32
- ▶ 2.2, “WSRR overview” on page 32
- ▶ 2.3, “Rational Asset Manager overview” on page 37
- ▶ 2.4, “WSRR Advanced Lifecycle Edition” on page 41
- ▶ 2.5, “Deployment topologies” on page 42
- ▶ 2.6, “Roles and responsibilities” on page 46
- ▶ 2.7, “Service governance assets” on page 48
- ▶ 2.8, “WSRR Advanced Lifecycle Edition synchronizable assets” on page 52
- ▶ 2.9, “Managing synchronization through the lifecycles” on page 60
- ▶ 2.10, “Additional WSRR-defined lifecycles” on page 68
- ▶ 2.11, “Additional resources” on page 71

## 2.1 Introduction

WSRR is the master metadata repository for service descriptions. It establishes a central point for finding and managing service metadata that is acquired from a number of sources. Rational Asset Manager is a collaborative software development asset management solution that gives organizations the ability to manage and govern the design, development, and consumption of services during design and development time.

Together, WSRR V6.3 and Rational Asset Manager V7.1 form WSRR Advanced Lifecycle Edition V6.3. This integrated product provides tight and consistent synchronization between the asset types, which can be accessed, viewed, and updated within each of the products.

**Important:** Although WSRR Advanced Lifecycle Edition V6.3 ships with Rational Asset Manager 7.1, Rational Asset Manager 7.1.1 is required for governance-based synchronization to take place.

In this chapter, we discuss how these products interact and federate with each other.

## 2.2 WSRR overview

There are many features and functions available with WSRR. This section provides the highlights of WSRR capabilities and introduces terminology that is used in later sections and chapters in this book.

WSRR is a Java 2 Platform, Enterprise Edition (J2EE) enterprise application, which runs on IBM WebSphere Application Server and uses a relational database as backup storage for all of the data that it stores. For WSRR V6.3, these databases can be DB2®, DB2 for z/OS, Oracle®, Microsoft® SQL Server®, and, for testing purposes, IBM Cloudscape.

### 2.2.1 Metadata and entities

There are two major types of user data stored in WSRR: registry data and repository data. *Registry* data can be characterized as the *metadata* attached to *repository* data loaded or created within WSRR. For example, when loading an XML document, not only can you access the document itself, but you can associate additional properties, classifications, and relationships (the metadata) to the *entity* created representing that XML document.

In addition to the physical document entities that are loaded into WSRR, generic entities can be created based on the classes defined in the *business model*<sup>1</sup> systems, which have been loaded by the administrator. Instances of these business model classes can, like physical document entities, have metadata associated with them. The WSRR default profile, the Governance Enablement Profile (GEP), defines separate business model classes, which can represent various concepts in an organization. For example, a Business Service can represent a high-level depiction of a service that has been, or needs to be, developed or deployed within an organization. Figure 2-1 shows the business model creation wizard view or Web user interface (UI) when creating an instance of this type of entity.

Targets	Status
<b>Credit Checking Service</b>   Add Relationship	✓ New
↳ Charter	CreditCheckingReq.doc   Replace   Remove ✓ New
↳ Service Interface Versions	Add Service Interface Specification
↳ Versions	Add Capability Version
↳ Dependency	Add Asset
↳ Owning Organization	Add Organization
↳ Artifacts	Add Document

Details

Edit Classifications

General Properties

\*Name

Credit Checking Service

Description

This business service will check whether a specified customer has sufficient credit to purchase a product from our web-site.

Business Requirements Link

Relationships

Charter

None

Service Interface Versions

None

Versions

None

Dependency

None

Figure 2-1 Creating a new Business Service business model instance in the Web UI

<sup>1</sup> Business models are collections of system and user-defined entities that are used to represent important business concepts in WSRR. Go to this Web site:  
[http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/cwsr\\_overview\\_whatsnew\\_busmodel.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/cwsr_overview_whatsnew_busmodel.html)

## 2.2.2 Assets

Classes defined in a business model system can be hierarchical, that is, one class can inherit the attributes of a super-class. The *Asset* class in WSRR is one of those super-class entities. A set of classes is defined in the default profile, and these classes inherit the default profile's properties and relationships. The classes are the types of entities that can be synchronized to and from Rational Asset Manager. Figure 2-2 shows the hierarchical nature of these synchronizable assets.

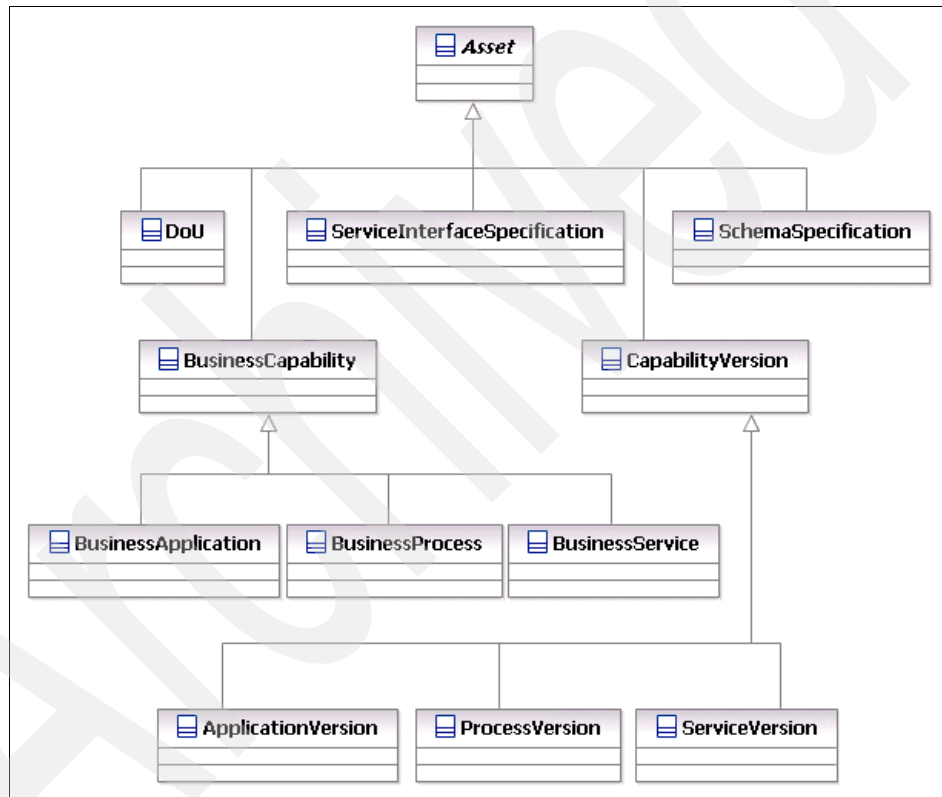


Figure 2-2 Asset hierarchy of synchronizable entities

The WSRR Advanced Lifecycle Edition provides a business domain classification system in WSRR that corresponds to the business domain categorization system in Rational Asset Manager. Applying these categories to the Assets in WSRR or Rational Asset Manager is a manual process and not

performed by the synchronization process. However, Figure 2-3 provides a visual representation of the hierarchy of these categories.

Business Domains	
<b>Business Administration</b> <ul style="list-style-type: none"><li>• Business Planning</li><li>• Staff Appraisals</li><li>• Account Administration</li><li>• Purchasing</li><li>• Product Administration</li><li>• Branch and Store Operations</li><li>• Business Unit Tracking</li></ul>	<b>Service and Sales</b> <ul style="list-style-type: none"><li>• Sales Planning</li><li>• Sales Management</li><li>• Sales</li><li>• Customer Service</li><li>• Collection</li></ul>
<b>New Business Development</b> <ul style="list-style-type: none"><li>• Sector Management</li><li>• Product Management</li><li>• Product Directory</li><li>• Marketing Campaigns</li><li>• Sector Planning</li></ul>	<b>Relationship Management</b> <ul style="list-style-type: none"><li>• Account Planning</li><li>• Relationship Management</li><li>• Credit Assessment</li><li>• Credit Administration</li></ul>
<b>Fulfillment</b> <ul style="list-style-type: none"><li>• Fulfillment Planning</li><li>• Fulfillment Monitoring</li><li>• Product Fulfillment</li><li>• Document Management</li></ul>	<b>Financial Control and Accounting</b> <ul style="list-style-type: none"><li>• Portfolio Planning</li><li>• Compliance</li><li>• Reconciliation</li><li>• Customer Accounts</li><li>• General Ledger</li></ul>

Figure 2-3 Business model domain hierarchy available in WSRR Advanced Lifecycle Edition

These categories represent different areas within an organization and can be associated with Assets in Rational Asset Manager and in WSRR. Individual or multiple categories can be tagged to existing or new Assets that will help group the Assets and will assist in searching Assets to enable reuse.

### 2.2.3 Governance capabilities

WSRR provides a variety of governance capabilities. Any piece of user data that is represented in WSRR can be governed and controlled within a user-definable lifecycle. A *lifecycle* is a set of *states* with *transitions* defined between them. Governance controls can be set up in WSRR to ensure that the correct person is performing the correct action to the correct entity at the correct time. You can see the key lifecycles in 2.9, “Managing synchronization through the lifecycles” on page 60 and 2.10, “Additional WSRR-defined lifecycles” on page 68. The WSRR default profile provides a set of lifecycles and governance control policies to enable the governance and control of entities defined within WSRR. We describe

the usage of this profile in detail in Part 2, “Top-down scenario” on page 95 and Part 3, “Other scenarios” on page 273.

## 2.2.4 Security

WSRR security has two levels:

- ▶ J2EE role mappings distinguish between WSRR Administrators and Users.
- ▶ Fine-grained access control, which is defined by the WSRR Administrator, specifies the WSRR roles with their mappings onto WebSphere Application Server user-registry principals: users and roles. We describe the WSRR default profile roles in 2.6, “Roles and responsibilities” on page 46. Their default mappings map to All Authenticated users, which you need to change after installation.

## 2.2.5 Web interface

You can administer WSRR and load, retrieve, delete, and govern data by using various interaction methods:

- ▶ The Web User Interface (WebUI)
- ▶ An Eclipse™ plug-in
- ▶ The programmatic APIs: Java Management Extensions (JMX)-based, Enterprise JavaBeans (EJB™), Web services, and Representational State Transfer (REST)
- ▶ The command line interface

We interact with WSRR in Part 2, “Top-down scenario” on page 95 and Part 3, “Other scenarios” on page 273 of this book by using the WebUI.

A WSRR perspective is a configured view of the data held within WSRR. Users in various WSRR roles have access to separate WSRR perspectives after they log in to the WSRR WebUI. An administrator has access to the full range of perspectives.

Figure 2-4 on page 37 shows the list of perspectives, with the development perspective currently selected. A user in the WSRR Development role is presented with either the Development perspective or the General User perspective.



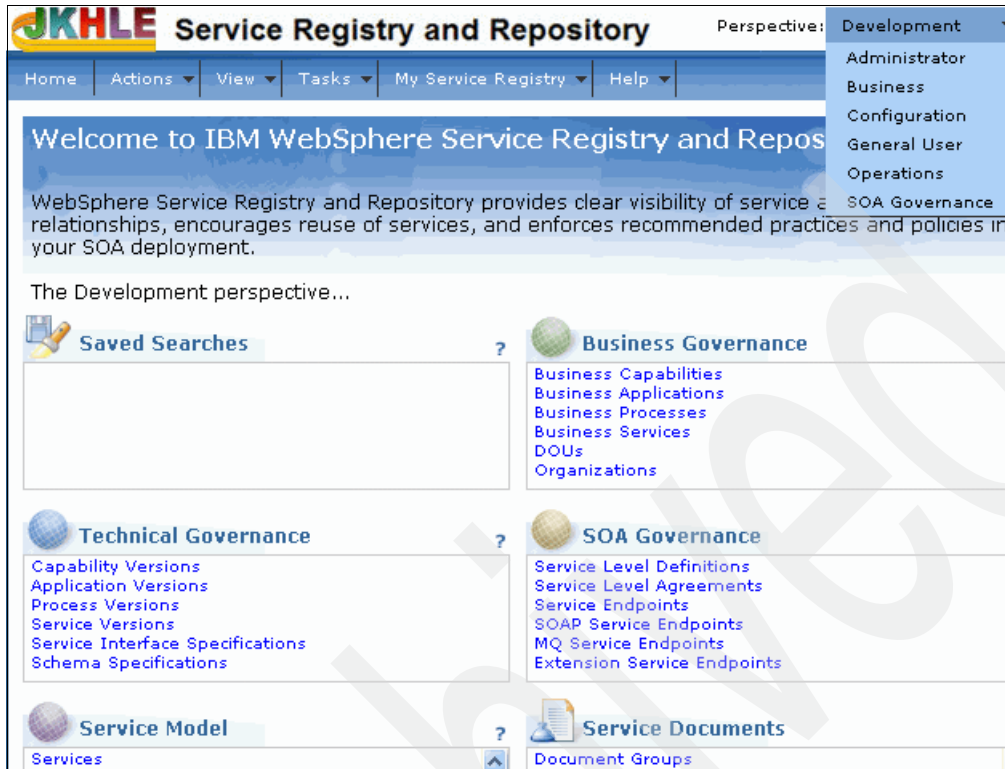


Figure 2-4 Development perspective with the Perspective selection menu opened

## 2.2.6 Integration with products

Additionally, IBM and non-IBM products have integration points with WSRR so that data can be published and retrieved from their environments. For example, WebSphere Message Broker has message nodes that can retrieve data from WSRR to influence the routing or transformation of the message through the defined message flow. You can obtain more information about integration with other products in the references listed in 2.11, “Additional resources” on page 71.

## 2.3 Rational Asset Manager overview

Users in development roles tend to use Rational Asset Manager more than WSRR. However, during our scenarios, we show that business and governance roles need to use Rational Asset Manager to create and approve business solutions and releases. This section provides the highlights of Rational Asset

Manager's capabilities and introduces terminology that is used in later sections and chapters in this book. The Rational Asset Manager information center describes all aspects of the product in detail.

Rational Asset Manager is a J2EE enterprise application that runs on IBM WebSphere Application Server. It uses a relational database as a backing store for all the data that it stores. The database can be DB2, Oracle, or Microsoft SQL Server for Rational Asset Manager V7.1.1.

### 2.3.1 Web interface

You can administer Rational Asset Manager and load, retrieve, delete, and govern data by using the Web user interface, an Eclipse plug-in, the programmatic Java APIs, and Apache ANT scripting. The interactions that we use in the scenarios in this book are through the Web user interface. Figure 2-5 shows the home page of Rational Asset Manager V7.1.1 where assets can be submitted and searched.



Figure 2-5 Rational Asset Manager V7.1.1 home page

## 2.3.2 Assets

User data is stored in Rational Asset Manager as *assets*, which are representations of specific *asset types*. Asset types are the primary level of organization within the repository. They facilitate governance and searching and are representative of the Asset business model class in WSRR. The WSRR Advanced Lifecycle Edition synchronization maps a subset of the Rational Asset Manager asset types to WSRR entities, which subclassifies the Asset business model class.

Submitted assets in Rational Asset Manager can have differing information and artifacts defined based on their type. New asset types can be defined by the administrator or loaded via an asset type library.

**Optional task:** It is possible for the administrator to create a Rational Asset Manager validator to ensure that Specification, Business Solution, and Release assets have an appropriate categorization. However, we do not discuss this capability in this book.

### Asset constraints

There are four major constraints that can be associated with an asset:

- ▶ Artifacts

A submitted asset can have associated documents constrained by format, label, or extension.

- ▶ Categories

A submitted asset can have categorizations applied to it. These categorizations are similar to classifications in WSRR. Categorizations are hierarchical. For example, the following categorization represents a Service in Service-Oriented Architecture (SOA) for Software Development:

Software Development → SOA → Service

- ▶ Relationships

A submitted asset can have relationships to other Rational Asset Manager assets.

- ▶ Attributes

A submitted asset can specify custom descriptors. These custom descriptors are similar to properties in WSRR. These types of attributes are name-value-pairs; they have a key name and an associated value.

## WSRR Advanced Lifecycle Edition Assets

The WSRR Advanced Lifecycle Edition library that is loaded into Rational Asset Manager allows Asset types to be synchronized with WSRR. These types are represented in Figure 2-6 along with their relevant categorizations.

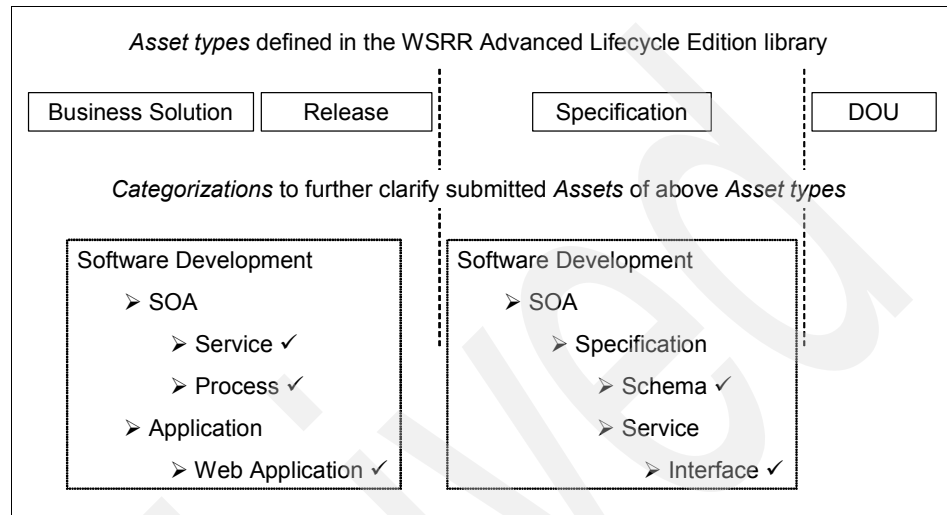


Figure 2-6 Asset types defined in the Advanced Lifecycle Edition library

Refer to 2.8, “WSRR Advanced Lifecycle Edition synchronizable assets” on page 52 for a description of the mappings of the entities that are represented in Figure 2-2 on page 34 and the entities in Figure 2-6.

### 2.3.3 Communities

*Communities* are collections of assets that are grouped by a common use and purpose. The assets in a community can have separate types, categories, relationships, and review processes. A community is an environment in which users interact with a group of related assets.

### 2.3.4 Review process

A *review process* can be set up for an asset if the asset can benefit from reviewer feedback. The review process can be initiated if a submitted asset to a community meets specific requirements: it is a particular asset type, it is in a specific categorization, or a combination of both. This process ensures that an asset is complete, accurate, and optimized for reuse.

## 2.4 WSRR Advanced Lifecycle Edition

WSRR Advanced Lifecycle Edition is the software solution that consists of both WSRR (see 2.2, “WSRR overview” on page 32) and Rational Asset Manager (see 2.3, “Rational Asset Manager overview” on page 37). This design, development, deployment, and runtime solution targets the service development and service deployment phases of the SOA service lifecycle, governing services from creation to consumption.

Synchronizing assets between WSRR and Rational Asset Manager, using WSRR Advanced Lifecycle Edition, provides the following benefits:

- ▶ Integrated repositories to store, catalog, and flexibly organize assets and services
- ▶ Customizable workflow for approvals and governance lifecycle
- ▶ Flexible solution for capturing relationships between services and support for impact analysis
- ▶ Ability to report on key metrics to support planning and assessment
- ▶ Integration with your SOA design and runtime systems, including other existing repositories

### 2.4.1 Synchronization

When an entity in one system (WSRR or Rational Asset Manager) is moved through specific transitions from one lifecycle state into another lifecycle state, changes to that entity are reflected in the other system. This approach provides a federated view of the data held in one or the other of the products. This federated view increases the traceability of where, when, and who performed what action and where the responsibility lies if fixes are required.

Synchronization using WSRR Advanced Lifecycle Edition is relevant for entities of type Asset represented in both WSRR and Rational Asset Manager:

- ▶ Asset types in WSRR represent those entities that can be synchronized with Rational Asset Manager. Asset types in WSRR, because they are WSRR entities, can be annotated with additional properties, classifications, and relationships to other entities. The definition of the Asset business model class in WSRR is *abstract* and has, in the Governance Enablement Profile, five subclasses: Business Capability, Capability Version, Service Interface Specification, Schema Specification, and Document of Understanding. We describe these service lifecycle Asset types in 2.7, “Service governance assets” on page 48.

- ▶ Asset types in Rational Asset Manager represent reusable entities to help solve specific business problems. They can be related to other Asset types, annotated, and categorized. You can load additional libraries into Rational Asset Manager so that its users can interact with specific instantiable Asset types (for example, Business Solution) as opposed to the abstract Asset type.

WSRR Advanced Lifecycle Edition provides an asset type library, which must be loaded into Rational Asset Manager to support the synchronization of these assets:

[http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/twsr\\_ale\\_install\\_ram\\_asset.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/twsr_ale_install_ram_asset.html)

## 2.5 Deployment topologies

The installation of WSRR Advanced Lifecycle Edition requires the installation of WSRR and Rational Asset Manager. They are both WebSphere Application Server enterprise applications and therefore take advantage of the application server's enterprise-ready capabilities, including scalability, security, and availability.

WSRR and Rational Asset Manager can coexist on the same WebSphere Application Server, which is useful for initial proof of concept, demonstration, or prototyping purposes. However, we recommend that you install WSRR and Rational Asset Manager in separate application service installation environments. Because their interaction patterns are for runtime integration and development activities, they need to be on separate partitions or hardware in scalable WebSphere Application Server installation environments.

This section describes a typical WSRR and Rational Asset Manager installation topology and where to find the relevant information. After we describe the topologies, we present a high-level architectural overview of the WSRR and Rational Asset Manager interaction, including the deployment architecture used by the scenarios in this book.

### 2.5.1 Typical WSRR installation topology

The number of registries that you deploy, and the ways in which they are configured, depends on the nature of your SOA environment and the goals of your WSRR deployment project.

While a variety of factors come into play when considering the appropriate topology for a WebSphere deployment, the primary factors to plan for typically include:

- ▶ Security
- ▶ Performance
- ▶ Throughput
- ▶ Scalability
- ▶ Availability
- ▶ Maintainability
- ▶ Session state

A recommended full production deployment consists of multiple registries to support the various stages of the service development and testing lifecycle. You can keep the service metadata for your testing and staging environments in separate registries to mirror more effectively what will happen when a service goes into production (refer to Figure 2-7).

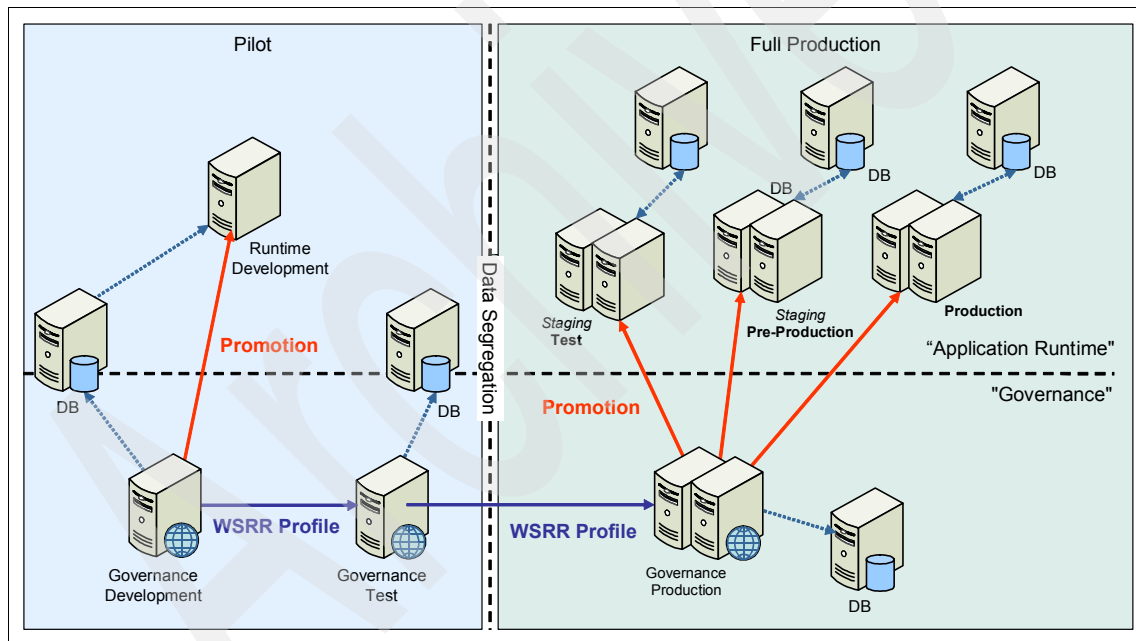


Figure 2-7 Recommended full production topology

Figure 2-7 shows a recommended production environment for WSRR. For more details of WSRR installation topologies, refer to the WSRR Information Center:

[http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/rwsr\\_deployment\\_topologies.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/rwsr_deployment_topologies.html)

## 2.5.2 Typical Rational Asset Manager installation topology

Rational Asset Manager's typical topology takes full advantage of a WebSphere Application Server clustered environment. For production, we advise that you create a front end for the cluster with resilient hardware and software to minimize failover impact: load balancer, proxy caches, and Web servers. The back-end databases (server and Asset repository) need to be RAID configured (refer to Figure 2-8).

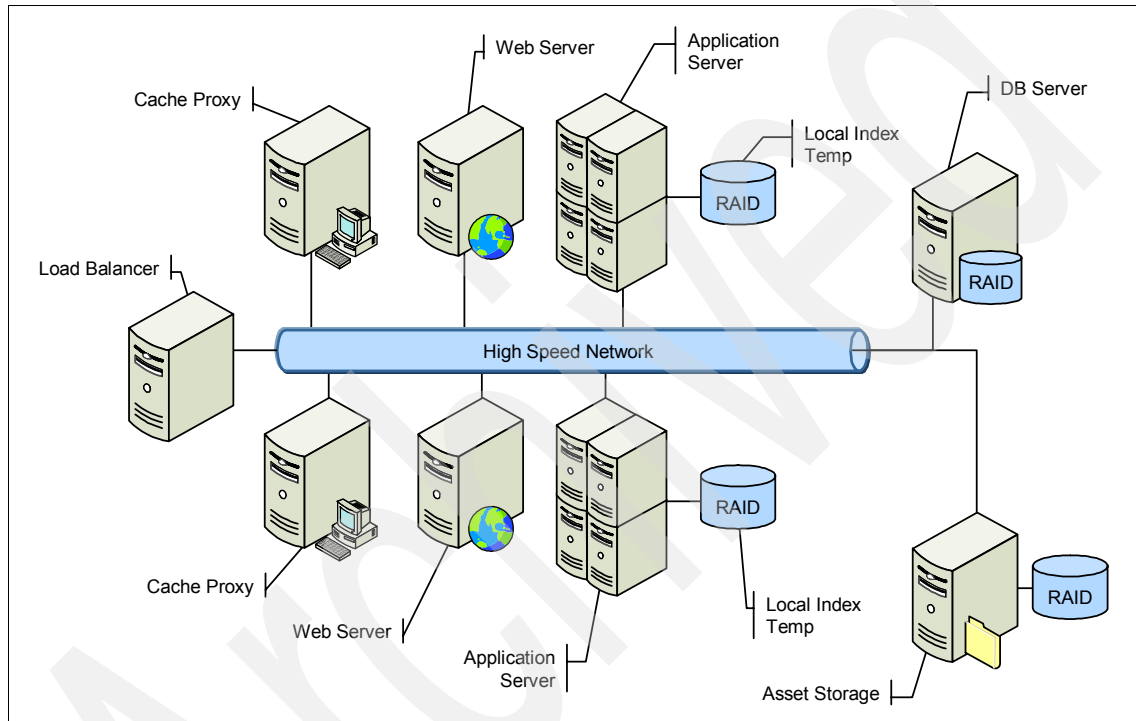


Figure 2-8 Typical Rational Asset Manager production deployment

*The Capacity Planning and Configuration Guide* describes in detail how to configure the Rational Asset Manager environment, taking into account scalability, availability, maintainability, security, and the expected number of assets:

[http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/RAM\\_plan\\_config\\_guide\\_v4.0.5.pdf](http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/RAM_plan_config_guide_v4.0.5.pdf)



## 2.5.3 Architectural deployment of WSRR Advanced Lifecycle Edition

WSRR Advanced Lifecycle Edition synchronizes entities between the WSRR deployment and the Rational Asset Manager deployment.

The following components in WSRR enable this integration:

- ▶ Scheduler setting
- ▶ Asset type mapping configuration (Advanced Lifecycle Edition Configuration)
- ▶ The Rational Asset Manager client JAR

The following components in Rational Asset Manager enable this integration:

- ▶ An additional Asset type library
- ▶ A new community connection
- ▶ A custom policy extension (Policy Governor plug-in)

Figure 2-9 shows these components and also shows the high-level components that are used during the synchronization. The following WSRR V6.3 Information Center article describes the steps that are required to install and configure these components:

[http://publib.boulder.ibm.com/infocenter/sr/v6r3/topic/com.ibm.sr.doc/t\\_wsr\\_ale\\_install.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/topic/com.ibm.sr.doc/t_wsr_ale_install.html)

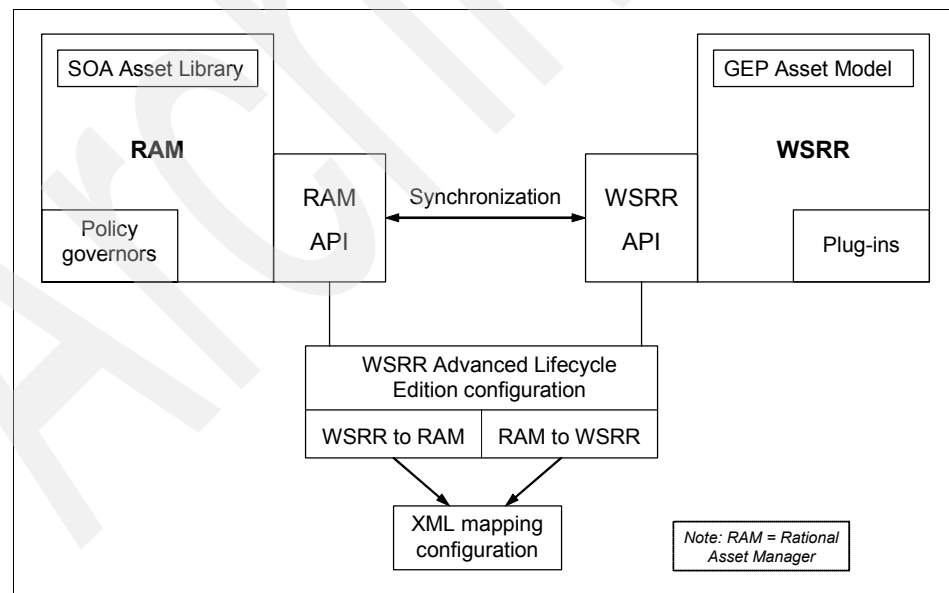


Figure 2-9 Advanced Lifecycle Edition asset synchronization overview

As lifecycle events occur to objects in either system, a queue of pending changes builds. At periodic intervals, a scheduled task is run in WSRR that examines the queue and updates the objects in one or the other (or both) systems to synchronize the metadata to maintain a consistent state. The WSRR Advanced Lifecycle administrator configures the frequency of this scheduled task. The frequency depends on the activity within the systems.

**Tip:** We recommend a task that runs every half hour. It is not the intention that the two systems be kept in step for every small change made to every object.

## 2.6 Roles and responsibilities

WSRR's Governance Enablement Profile (GEP) provides a number of roles for use with WSRR Advanced Lifecycle Edition. The roles defined in the profile are listed here:

- ▶ GEPAdministrator
- ▶ GEPUser
- ▶ Business
- ▶ SOAGovernance
- ▶ Development
- ▶ Operations

You assign a user to an appropriate role according to the SOA-related activities for which that user is responsible.

We describe the Development role, where crossover occurs between WSRR and Rational Asset Manager, in more detail in 2.6.5, "Development" on page 47.

The following sections describe the main roles defined within the Governance Enablement Profile.

### 2.6.1 GEPAdministrator

Users within the GEPAdministrator role have access to the Administrator perspective<sup>2</sup> that will provide the ability to update entities using consistent views for all the business models that have been defined. This role is normally configured to be able to import and export data from WSRR.

<sup>2</sup> There is also a Configuration perspective, which anyone in the WSRR J2EE Administrator role is able to see. For more details, see the Web site:  
[http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/twsr\\_configrn\\_userroles08.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/twsr_configrn_userroles08.html)

## 2.6.2 GEPUser

The GEPUser is, by default, for any authenticated user. The GEPUser role allows you to retrieve and view entities created by you or those entities that are in an Approved lifecycle state.

## 2.6.3 Business

The Business role represents business managers, analysts, and subject matter experts, who are interested in how the SOA services and processes contribute to the business. The users in this role define the requirements and the business capabilities that are required by their organization. They have the responsibility for ensuring that the SOA solutions that are delivered meet the business' need.

## 2.6.4 SOAGovernance

The SOAGovernance role represents architects, architecture boards, and SOA Centers of Excellence (CoE). This role also includes individuals from other roles (business, development, and operations) who are responsible for defining the governance processes, policies, and standards that are shared across all organizations to ensure the effective interoperability, agility, and robustness of the SOA solutions.

## 2.6.5 Development

The Development role represents software development practitioners, including architects, release managers, software developers, testers, assembly developers (who use tools, such as WebSphere Integration Developer), integrators, and asset librarians. The users in this role develop the software specifications and implementations to realize the requirements provided by the business, and they are responsible for ensuring that their implementation meets the business needs and adheres to the governance standards. Development roles are usually associated with a specific organization or department, with responsibility for delivering implementations to support a particular area of the business.

## 2.6.6 Operations

The Operations role represents operations managers, operations architects, system administrators, integration testers, and IT resource managers. The users in this role manage the IT infrastructure by deploying, configuring, and testing the implementations produced by development. They are responsible for operational

quality of service and capacity planning. Although their major activities occur later in the lifecycle, they can provide operational viewpoints during all specification and review activities to ensure that the planned capabilities can be successfully delivered. Operations roles can be centralized or arranged by line of business (LoB) or organization, according to individual company preferences.

## 2.7 Service governance assets

The service governance lifecycle that is mentioned in 1.3.3, “Service governance within SOA governance” on page 13 concerns service-related entities that are controlled in the model, assemble, deploy, and manage phases. Those service-related entities are relevant to separate roles at specific stages in the lifecycle.

Figure 2-10 on page 49 demonstrates that there are numerous service-related entities that are important throughout the service lifecycle for the roles that are described in 2.6, “Roles and responsibilities” on page 46. Figure 2-10 on page 49 shows those entities and Assets that can be found in each part of WSRR Advanced Lifecycle Edition. Rational Asset Manager’s primary focus is on development time activities whereas WSRR’s primary focus is on service governance and runtime activities. However, these activities overlap and even though one type of Asset might be defined in one environment, it can be represented in the other environments to aid governance and control. For example, Shared Components is only in Rational Asset Manager, but Service Interface Specifications is in both WSRR and Rational Asset Manager.

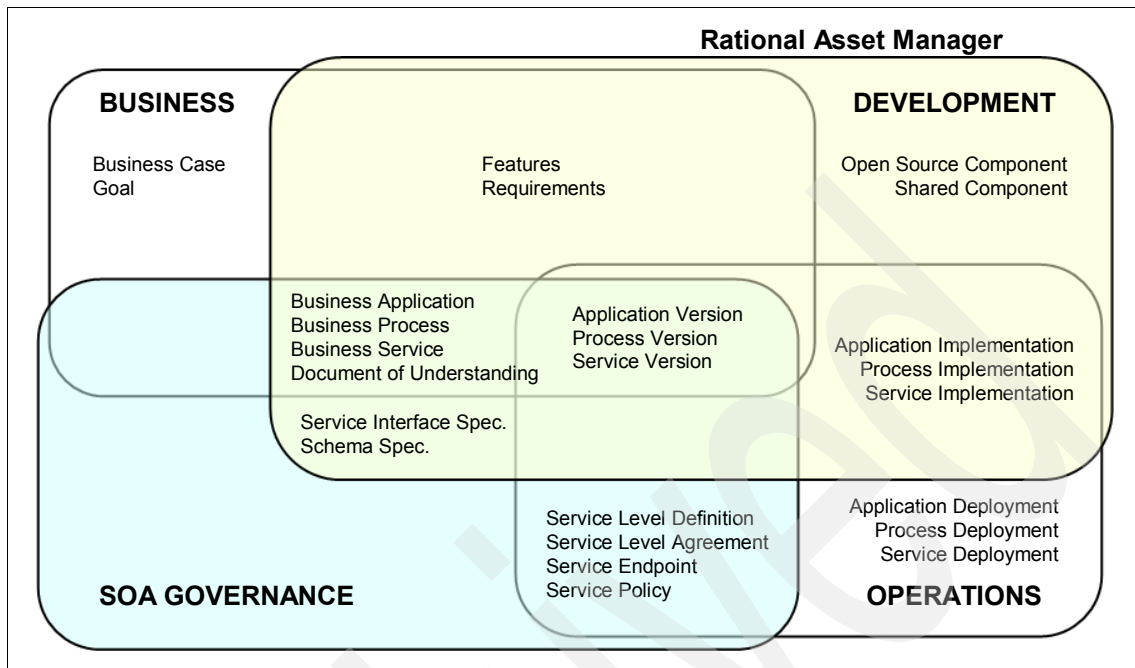


Figure 2-10 Service governance assets

## 2.7.1 Entities for the Business role

Users in the Business role are concerned with the following entities:

- ▶ **Business Cases**  
Justification for spending money on a particular activity or developing a particular service.
- ▶ **Goals**  
What returns are expected from the delivered service. These goals can be merged with business cases.
- ▶ **Features**  
What features and functionality must exist when a service is delivered.
- ▶ **Requirements**  
Those functional and non-functional requirements that need to be delivered as part of a current or future release of a service.

- ▶ **Business Capabilities: Applications, Processes, and Services**  
Abstraction from the business case to start traceability (in WSRR and Rational Asset Manager) when starting to think about developing a service. Also, see 2.8.1, “Business Capability, Business Solution” on page 55.
- ▶ **Documents of understanding (DOUs)**  
Linkage between two parts of an organization where one part is relying on another part to provide a particular service. For example, Finance-IT might rely on Common Services-IT to deliver widget X that needs to be used in their widget: the consumer and provider relationship. Also, see 2.8.5, “Document of Understanding” on page 60.
- ▶ **Capability Versions: Applications, Processes, and Services**  
A realization of a Business Capability describing what will be delivered in a particular version. For example, one version of a Business Capability might deliver features A, B, and C of the original Business Capability’s A to Z of features. Also, see 2.8.2, “Capability Version, Release” on page 57.

## 2.7.2 Entities for the Development role

Users in the Development role are concerned with the following entities that are used by the Business role:

- ▶ Features
- ▶ Requirements
- ▶ Business Capabilities: Applications, Processes, and Services
- ▶ DOUs
- ▶ Capability Versions: Applications, Processes, and Services

In addition, users in the Development role are concerned with these components:

- ▶ **Open Source Components**  
Those pieces of development that come from Open Source sources.
- ▶ **Shared Components**  
Those pieces of development owned by the Development organization.
- ▶ **Implementations: Applications, Processes, and Services**  
Delivered implementations of a service.
- ▶ **Specifications: Service Interfaces and Schemas**  
These specifications are reference guides that describe exactly how a message must be defined. These specifications allow the development team to know exactly what messages it can expect from the invokers of the service

and exactly what messages to return. The difference between Service Interface Specifications (SIS) and Schema Specifications (SS) is that the SIS describes the operations with parameters and return types and the SS describes the format of the parameters and return types. Also, see 2.8.3, “Service Interface Specification, Specification” on page 58 and 2.8.4, “Schema Specification, Specification” on page 59.

### 2.7.3 Entities for the SOA Governance role

The SOA Governance role is also concerned with the following entities that have previously been discussed:

- ▶ Business Capabilities: Applications, Processes, and Services
- ▶ DOUs
- ▶ Capability Versions: Applications, Processes, and Services
- ▶ Specifications: Service Interfaces and Schemas

In addition, the SOA Governance role is concerned with these components:

- ▶ Service Level Definitions (SLD)  
Formal specification of the physical communication mechanisms used to deliver the messages for interaction with a provided service and includes quality of service characteristics related to the interaction.
- ▶ Service Level Agreements (SLA)  
Define a specific dependency that a consuming service has on a specific SLD of a provider's service. For example, if the SLD defines that its service can support a maximum of 100 transactions per second (tps), the consuming service might only be allowed a maximum of 10 tps as defined in the Service Level Agreement.
- ▶ Service Endpoints  
The real endpoints where a service has been deployed. A specific SLD will have a number of defined Service Endpoints. An example of a Service Endpoint is the SOAPAddress element (with location attribute) in a Web Services Description Language (WSDL) file.
- ▶ Service Policies  
Policies that can control how a service needs to be invoked. These policies can be WebSphere Policies (WS-Policies) restricting access to a particular security protocol.

## 2.7.4 Entities for the Operations role

The Operations role is concerned with the following entities that have been previously discussed:

- ▶ Capability Versions: Applications, Processes, and Services
- ▶ Implementations: Applications, Processes, and Services
- ▶ Service Level Definitions
- ▶ Service Level Agreements
- ▶ Service Endpoints
- ▶ Service Policies

In addition, the Operations role is concerned with the following entity:

- ▶ Deployments: Applications, Processes, and Services

Installation representations of where and how the relevant services are deployed. The services will be deployed on a particular server and will have a specific endpoint to allow those users (consumers) of the service to invoke it.

## 2.8 WSRR Advanced Lifecycle Edition synchronizable assets

The definition of the Asset business model class in WSRR is abstract. Any business model class that subclassifies it can be synchronized with Rational Asset Manager. The Asset type on its own has no meaning (which is why it is abstract), so by subclassifying it, semantics can be implied and defined.

There are five subclasses within the default WSRR Governance Enablement Profile (GEP):

- ▶ Business Capability
- ▶ Capability Version
- ▶ Service Interface Specification
- ▶ Schema Specification
- ▶ Document of Understanding

Within Rational Asset Manager, with the appropriate Advanced Lifecycle Edition library loaded, there are four matching entities:

- ▶ Business Solution
- ▶ Release
- ▶ Specification
- ▶ Document of Understanding



This subset of the entities described is synchronizable in WSRR Advanced Lifecycle Edition and is defined as Asset types:

- ▶ Business Capabilities: Applications, Processes, and Services  
See 2.8.1, “Business Capability, Business Solution” on page 55 for more details.
- ▶ DOUs  
See 2.8.5, “Document of Understanding” on page 60 for more details.
- ▶ Capability Versions: Applications, Processes, and Services  
See 2.8.2, “Capability Version, Release” on page 57 for more details.
- ▶ Specifications: Service Interfaces and Schemas  
See 2.8.3, “Service Interface Specification, Specification” on page 58 and 2.8.4, “Schema Specification, Specification” on page 59 for more details.

You can create these synchronizable entities in WSRR or Rational Asset Manager. When specific stages are reached in the service lifecycle, they will be synchronized into the complementary environment. Even though you can create entities in WSRR or Rational Asset Manager, the scenarios described in this book provide recommendations.

In this section, we describe these GEP Asset types in more detail and describe their significance and representation in and across WSRR Advanced Lifecycle Edition. Figure 2-11 on page 54 shows how the entities in the GEP are conceptually interrelated.

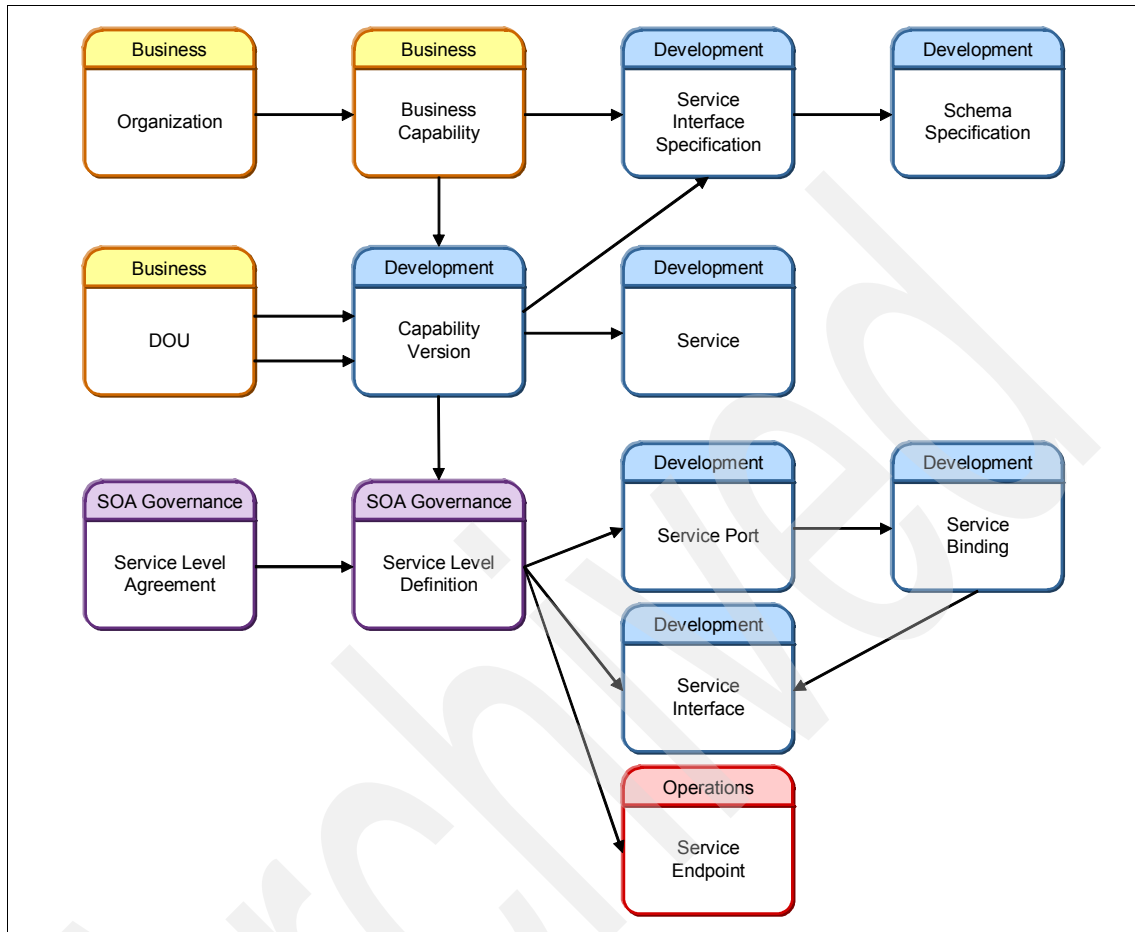


Figure 2-11 Interrelationship of entities that are representable in WSRR

There is a natural flow when creating the entities in Figure 2-11 in a top-down way. A top-down approach is where the business is driving the change; a new feature or capability is required by an organization. This top-down approach is the recommended practice and is described in 3.2, “Top-down development” on page 74 and in Part 2, “Top-down scenario” on page 95. This flow is an example of a top-down approach:

1. The Business role defines a new Business Capability that is required by the organization. They can create a relationship to the organization it applies to and who will own it.
2. The Development role can define a set of Specifications against the Business Capability, which can be used when developing the new service.

3. The Development role will realize and start developing a particular [Capability] version of the Business Capability and might optionally relate refined specifications.
4. The Development team will implement and deliver the service providing service entities based on the requirements, use cases, and other available materials.
5. The SOA Governance role will define, in conjunction with Development, what qualities of service the implementation's endpoint can deliver by defining the Service Level Definition.
6. Operations will deploy the service at a particular endpoint and attach it to the Service Level Definition.
7. If another capability version needs to consume this particular service's Service Level Definition, a Document of Understanding and a Service Level Agreement can be established defining exactly who and how it will be consumed.

This example flow shows that there are additional entities defined by the GEP, which will not get synchronized but can exist in Rational Asset Manager or in WSRR. These additional entities include entities that are created (or correlated) when WSDL and XSD documents are loaded, subscription representations (Service Level Agreements), and endpoint representations (Service Level Definitions and ServiceEndpoints). These entities are all important within the federated view of Services across Rational Asset Manager and WSRR, but they are not covered in this section.

All the entities that are defined in the GEP assist the runtime environments and WSRR users:

- ▶ Runtime environments or enterprise service buses (ESBs) with their determination as to whether a consumer is allowed to invoke a particular endpoint, whether an endpoint is available, or which endpoint to choose
- ▶ A WSRR user when navigating from the business-level description of a service (the Business Capability) through who is using it to where that service is really deployed

### 2.8.1 Business Capability, Business Solution

The Business Capability entity in WSRR (and the Business Solution representation in Rational Asset Manager) expresses a generalized capability within the organization. This generalized capability is what users in the Business role see as a solution or service offered by their company and can be considered the topmost entity and starting point for traceability. From the Business view of

services offered, a user can navigate to where it is really deployed to a runtime environment.

### **WSRR representation**

The Business Capability entity is a superclass for three types of solutions that can be offered and represented in WSRR:

- ▶ Business Service  
A realized or unrealized (depending on its position in the lifecycle) service in the organization
- ▶ Business Process  
A collection of related activities or tasks in an organization
- ▶ Business Application  
An application that represents a back-end business transaction exposed as a service or a particular channel to market, such as a Web or Portal application

### **Rational Asset Manager representation**

The equivalent entity when identifying a Business Capability in Rational Asset Manager is a Business Solution with a Categorization. The three Business Capabilities are represented in Rational Asset Manager using the following Categorizations:

- ▶ Business Solution  
Categorization: Software Development → SOA → Service
- ▶ Business Solution  
Categorization: Software Development → SOA → Process
- ▶ Business Solution  
Categorization: Software Development → Application → Web Application

### **Usage**

Business Capabilities (or Business Solutions) are used as the starting point for linking business requirements (Business view) with technical functionality (IT department) in the SOA environment.

Business users expect to see and are expected to view these entities in WSRR Advanced Lifecycle Edition (using either the WSRR WebUI or Rational Asset Manager UI), helping them understand what solutions are offered internally and externally by their organization. This capability can help them to make more informed strategic decisions based on the knowledge that they already have.

## 2.8.2 Capability Version, Release

The Capability Version entity in WSRR (and the Release representation in Rational Asset Manager) expresses a release and implementation (a realization) of the generalized Business Capability within the organization. The business user reviews the high-level description of the business solution and can then create or view a realization of that solution. The realization describes the capabilities that will be offered in this particular implementation/version.

### WSRR representation

The Capability Version entity is a super-class for three types of implementation that can be represented in WSRR:

- ▶ Service Version  
A specific version or release of a business service that provides a range of functional and nonfunctional specifications
- ▶ Process Version  
A specific version or release of an SOA process that can expose several of its capabilities as services with the appropriate Service Level Definitions
- ▶ Application Version  
A specific version or release of a Web application that only consumes services or an exposed business application

### Rational Asset Manager representation

The equivalent entity when identifying a Capability Version in Rational Asset Manager is a Release with a Categorization. The three Capability Versions are represented in Rational Asset Manager using the following Categorizations:

- ▶ Release  
Categorization: Software Development → SOA → Service
- ▶ Release  
Categorization: Software Development → SOA → Process
- ▶ Release  
Categorization: Software Development → Application → Web Application

### Usage

Users with the Development role are interested in using and defining Capability Versions (or Releases) to represent specific versions of a generalized Business Capability (or Business Solution). It is possible to navigate from this Capability Version in order to identify which, if any, other services it consumes using the

Service Level Agreement relationship named “consumes” and what level of service this version’s endpoints offer using the Service Level Definition relationship named “provides”.

The specified version on the Capability Version is significant. The recommended practice is that the specified version is in the format `major.minor`. This format indicates to the readers of the definitions whether the particular version of a Business Capability is compatible. Versions that do not have the same major version indicator are incompatible. Versions that differ by their minor version indicator only are backwardly compatible. For example, a message intended for a 1.0 version of a service can be targeted at a 1.1 version of a service. For more information about service version management, and compatible and incompatible version levels, refer to 3.4, “Service version management” on page 76.

### 2.8.3 Service Interface Specification, Specification

The Service Interface Specification entity in WSRR (and the Specification representation in Rational Asset Manager) expresses a particular interface (interaction pattern and message structure). This interface can be generic if attached to the Business Capability or more fine-grained (or specific) when attached to the Capability Version: the realized versions of the higher level Business Capability. The specification, at either level, does not deal with deployment or runtime specifics that are associated with the endpoint description of a solution (for example, its transport type or quality of service). Most commonly, the Service Interface Specification will be realized by a WSDL port-type entity (the Service Interface of a Web Service deployment).

SOA Governance roles are concerned with defining the Service Interface Specifications at the Business Capability level to ensure that it is compliant across the organization and complies with upcoming strategic decisions. Development roles then base the new implementation of the solution on the specification that is defined at the Business Capability level. Or, they can create/refine, in a relevant and compatible manner, a new Service Interface Specification, which is relevant for the implementation of the new Capability Version.

#### **WSRR representation**

The Service Interface Specification entity in WSRR is represented as a Service Interface Specification.

#### **Rational Asset Manager representation**

The equivalent entity when identifying a Service Interface Specification in Rational Asset Manager is a Specification with a Categorization. The Service

Interface Specification is represented in Rational Asset Manager using the following Categorization:

- Specification

Categorization: Software Development → SOA → Specification → Service → Interface

## Usage

You can define Service Interface Specifications at the Business Capability (Business Solution) level or the Capability Version (Release) level. At the Business Capability level, Service Interface Specifications specify a more general interface, which can then be refined for particular versions (Capability Versions) of that Business Capability.

Versioning of these Specifications is also important and recommended so that you can identify compatible (extended) and incompatible (changed) message structures.

## 2.8.4 Schema Specification, Specification

The Schema Specification entity in WSRR (and the Specification representation in Rational Asset Manager) expresses the formal description of the message structures necessary for the implementation of the Service Interface Specification. When developing a new solution that requires messages to flow from one location to another location, this formal representation is important so that consumer services and provider services know the format of messages that will be sent and received.

Development roles are concerned with the implementation details of defining a new Schema Specification, because the Development roles will need to send and receive messages within their implementations.

### WSRR representation

The Schema Specification entity in WSRR is represented as a Schema Specification.

### Rational Asset Manager representation

The equivalent entity when identifying a Schema Specification in Rational Asset Manager is a Specification with a Categorization. The Schema Specification is represented in Rational Asset Manager using the following Categorization:

- Specification

Categorization: Software Development → SOA → Specification → Schema

## 2.8.5 Document of Understanding

The DOU entity in WSRR (and the DOU representation in Rational Asset Manager) expresses an agreement between an implementation by a consuming organization and an implementation by a providing organization. The DOU relates the realized consuming implementation (one Capability Version) with the realized providing implementation (another Capability Version). When the implementations have been completed and the providing implementation is deployed, the DOU can establish a Service Level Agreement (subscription/contract). This Service Level Agreement provides particular properties describing how the consuming implementation can use the providing implementation at run time.

### **WSRR representation**

The Document of Understanding entity in WSRR is represented as a DOU.

### **Rational Asset Manager representation**

The equivalent entity when identifying a Document of Understanding in Rational Asset Manager is a DOU entity. The DOU entity does not require any further clarification in Rational Asset Manager using a Categorization.

### **Usage**

The Document of Understanding is important, because it identifies which implementation of a Business Capability will consume which other implementation of another Business Capability.

The DOU can be used to initiate a subscription request to control the consumption of the provided implementation (using the Service Level Agreement entity).

## 2.9 Managing synchronization through the lifecycles

The Governance Enablement Profile (GEP) Asset types that are described in 2.8, “WSRR Advanced Lifecycle Edition synchronizable assets” on page 52 can be synchronized within WSRR Advanced Lifecycle Edition. (Note that classifications are not synchronized). Therefore, entities of those types (which are created in either the WSRR or Rational Asset Manager systems) can be, as they are transitioned through their respective lifecycles, represented by a corresponding entity in the complementary system with matching metadata (attributes/properties and relationships).



When synchronization is enabled in WSRR Advanced Lifecycle Edition, the most appropriate environment can be used by a relevant role to perform various actions during the appropriate stages of the relevant lifecycles.

In the GEP, there are three lifecycles that are relevant for governing and synchronizing entities in WSRR Advanced Lifecycle Edition:

- ▶ Capability lifecycle
- ▶ SOA lifecycle
- ▶ Asset lifecycle

You can view the states within these lifecycles that are relevant during the synchronization points in Figure 2-12 on page 62, Figure 2-13 on page 63, and Figure 2-17 on page 67.

Rational Asset Manager also has a lifecycle for assets within its own environment. The Rational Asset Manager has three lifecycle states:

- ▶ Identified
- ▶ In Review
- ▶ Approved

These states can apply to any asset in Rational Asset Manager.

Synchronization occurs at specific stages in Rational Asset Manager or in one of the three WSRR lifecycles, depending on the asset type.

### 2.9.1 Capability lifecycle

The GEP Asset types that will be governed using the Capability lifecycle are all of the Business Capability entities: Business Application, Business Process, and Business Service. The Capability lifecycle that is available in WSRR is shown in Figure 2-12 on page 62 alongside the matching lifecycle states in Rational Asset Manager. For example, Business Solution assets in Draft in Rational Asset Manager have a corresponding Business Capability in the Capability Identified state in WSRR.

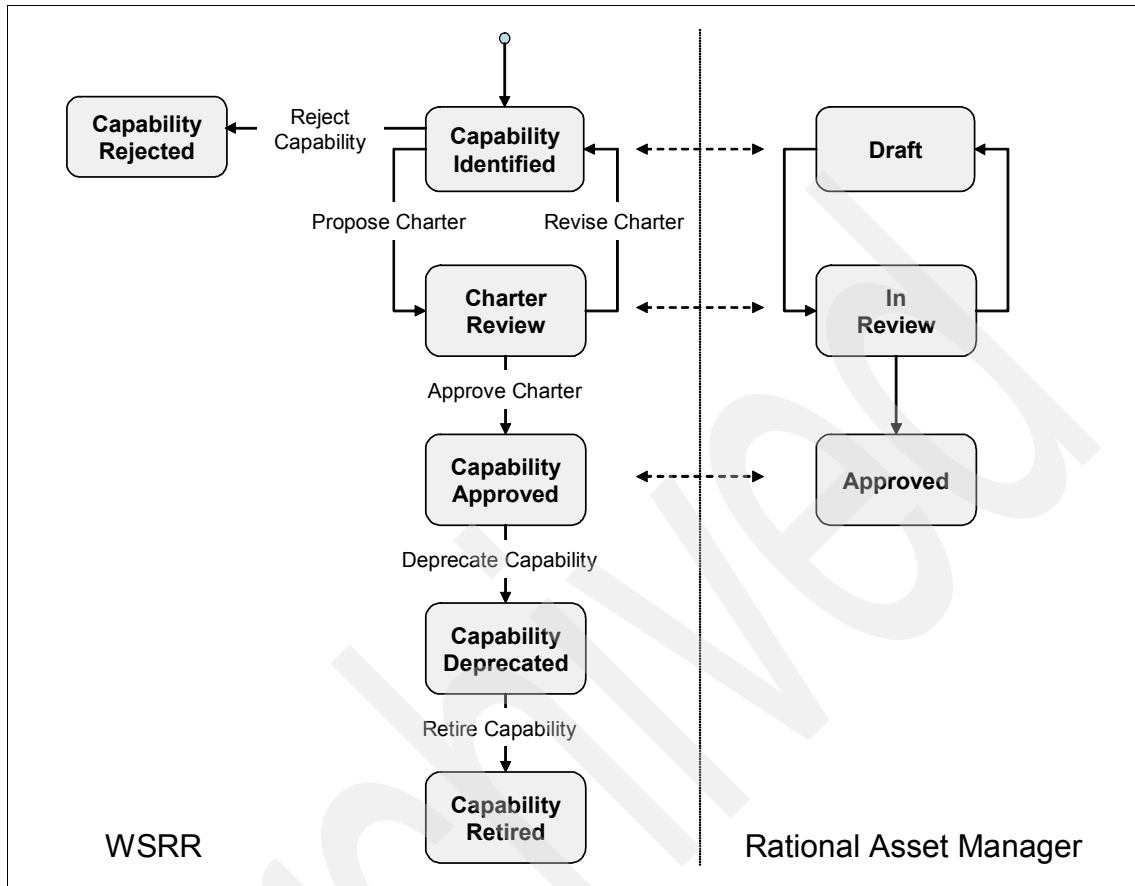


Figure 2-12 Capability lifecycle and synchronization with Rational Asset Manager

The recommended practice for objects in the Capability lifecycle is that those objects are created first in Rational Asset Manager. Entities created in WSRR, and therefore automatically created in Rational Asset Manager during synchronization, can only be placed into one Rational Asset Manager community.

Another limitation here is that a charter must be attached to the asset in WSRR so that it will synchronize correctly. Further development of the asset and its artifacts can then take place within the Rational Asset Manager environment, with changes reflected back to WSRR as the asset goes through the approval cycles in Rational Asset Manager. After the asset has been approved in Rational Asset Manager and any artifacts associated with it have been published back to WSRR, the lifecycle can continue in WSRR to approve the asset and deploy it to production.

## 2.9.2 SOA lifecycle

The GEP Asset types that will be governed using the SOA Lifecycle are all of the Capability Version entities: Application Version, Process Version, and Service Version.

### SOA lifecycle: Model phase

This lifecycle, which is available in WSRR, is shown in Figure 2-13 alongside the matching lifecycle states in Rational Asset Manager. For example, Release assets in Draft in Rational Asset Manager have a corresponding Capability Version in the Identified state in WSRR.

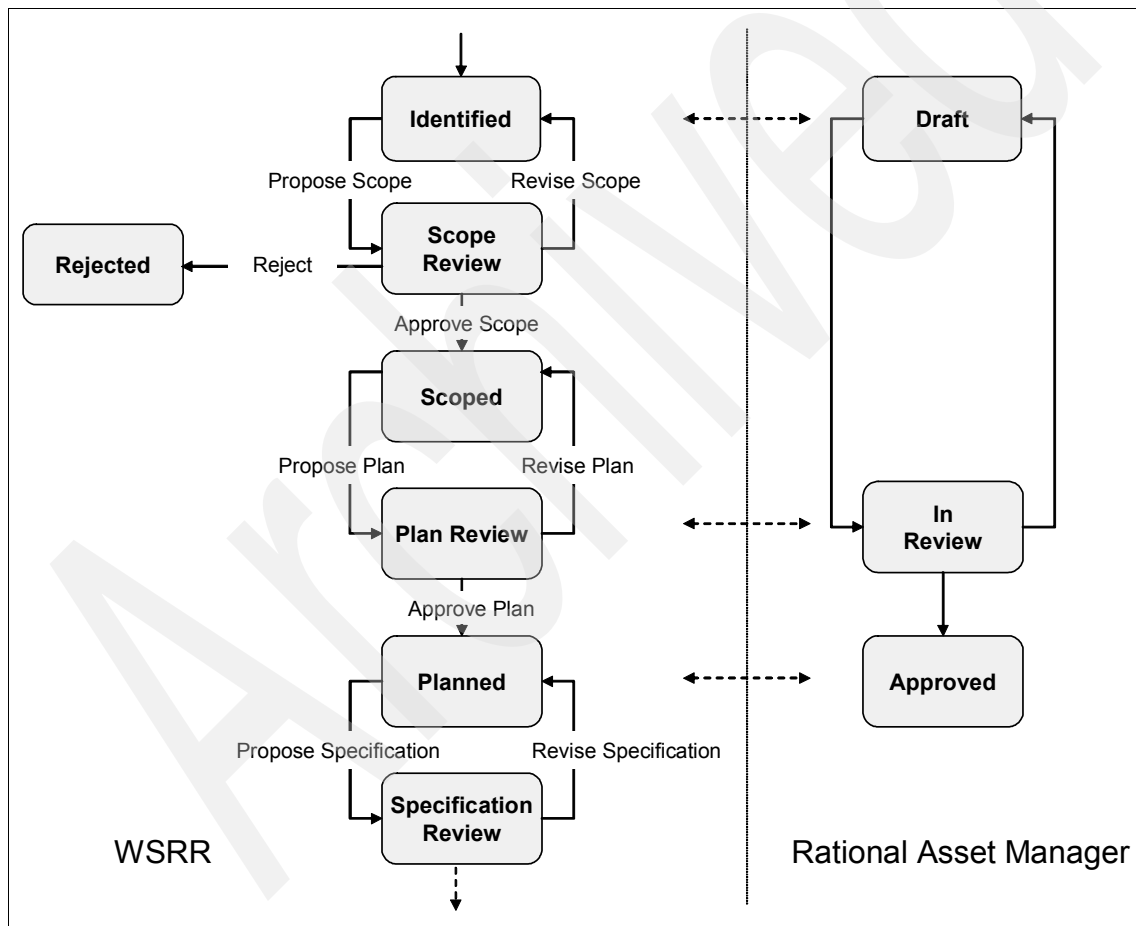


Figure 2-13 Model phase of the SOA lifecycle and synchronization with Rational Asset Manager

The recommended practice for objects in the SOA lifecycle is that those objects are created first in Rational Asset Manager. Entities that are created in WSRR and, therefore, that are automatically created in Rational Asset Manager during synchronization, can only be placed into one Rational Asset Manager community.

Further development of the asset and its artifacts can take place in WSRR and the Rational Asset Manager environment, with changes reflected within WSRR Advanced Lifecycle Edition. The asset goes through the approval cycles in Rational Asset Manager, and artifacts can be attached and reflected across the systems. The lifecycle can continue in WSRR to further plan, approve, and deploy the service to production.

This SOA lifecycle continues in WSRR into the assemble, deploy, and manage phases of the service lifecycle. We elaborate on these phases during the top-down scenario discussion in Part 2, “Top-down scenario” on page 95.

### SOA lifecycle: Assemble phase

Figure 2-14 shows the Assemble phase of the SOA lifecycle.

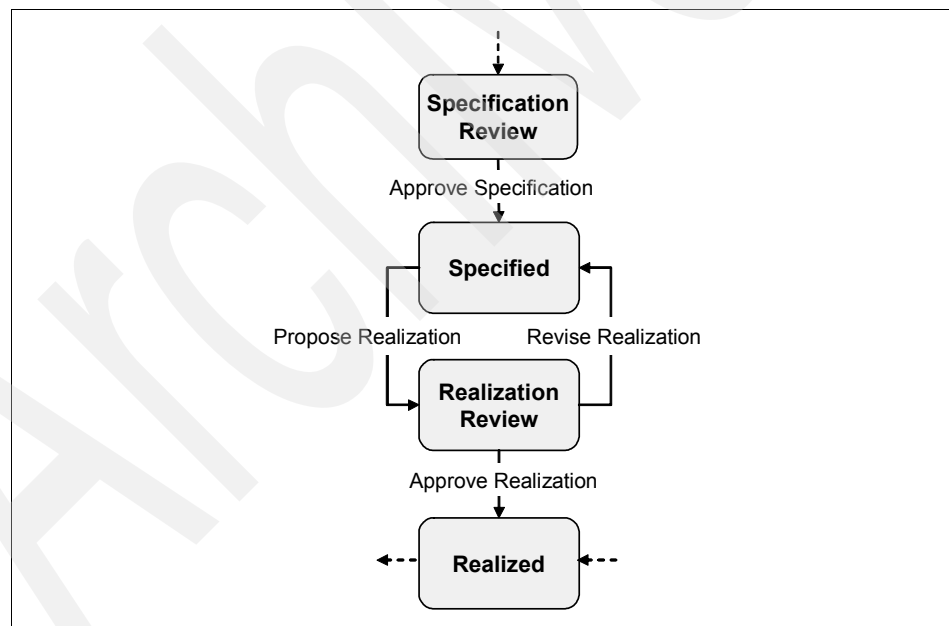


Figure 2-14 The assemble phase of the SOA lifecycle

### SOA lifecycle: Deploy phase

Figure 2-15 on page 65 shows the Deploy phase of the SOA lifecycle.

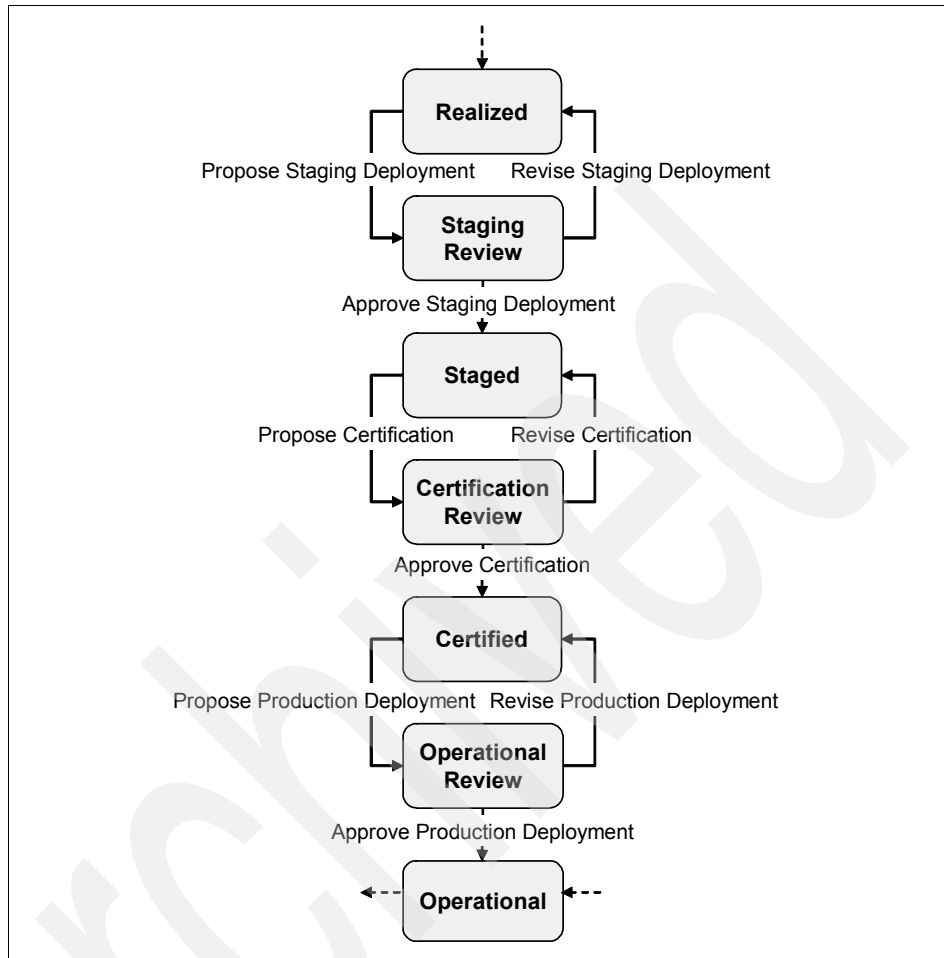


Figure 2-15 The deploy phase of the SOA lifecycle

## SOA lifecycle: Manage phase

Figure 2-16 on page 66 shows the Manage phase of the SOA lifecycle.

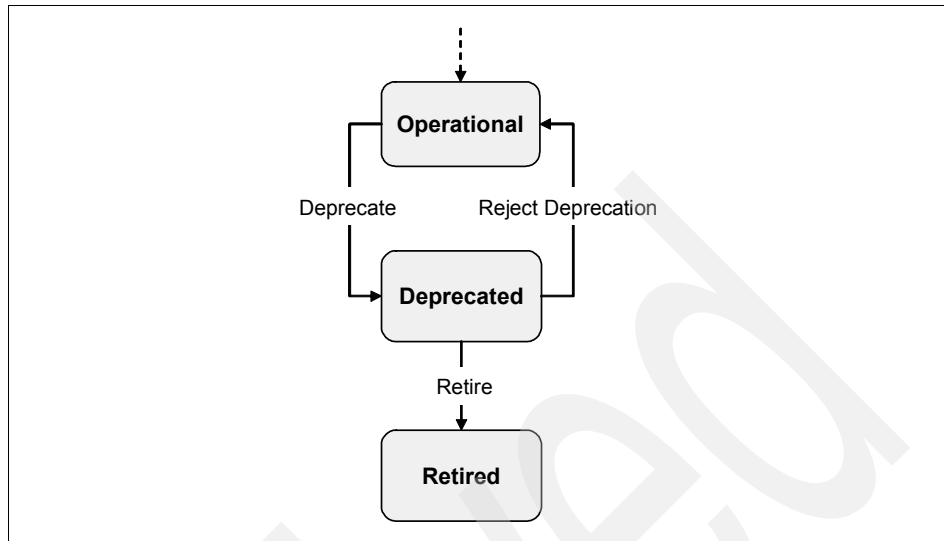


Figure 2-16 The manage phase of the SOA lifecycle

### 2.9.3 Asset lifecycle

The GEP Asset types that will be governed using the Asset Lifecycle are the Document of Understanding, Service Interface Specification, and Schema Specification entities. This lifecycle, which is available in WSRR, is shown in Figure 2-17 on page 67 alongside the matching lifecycle states in Rational Asset Manager. For example, a Document of Understanding asset in Draft in Rational Asset Manager has a corresponding Document of Understanding in the Asset Identified state in WSRR.

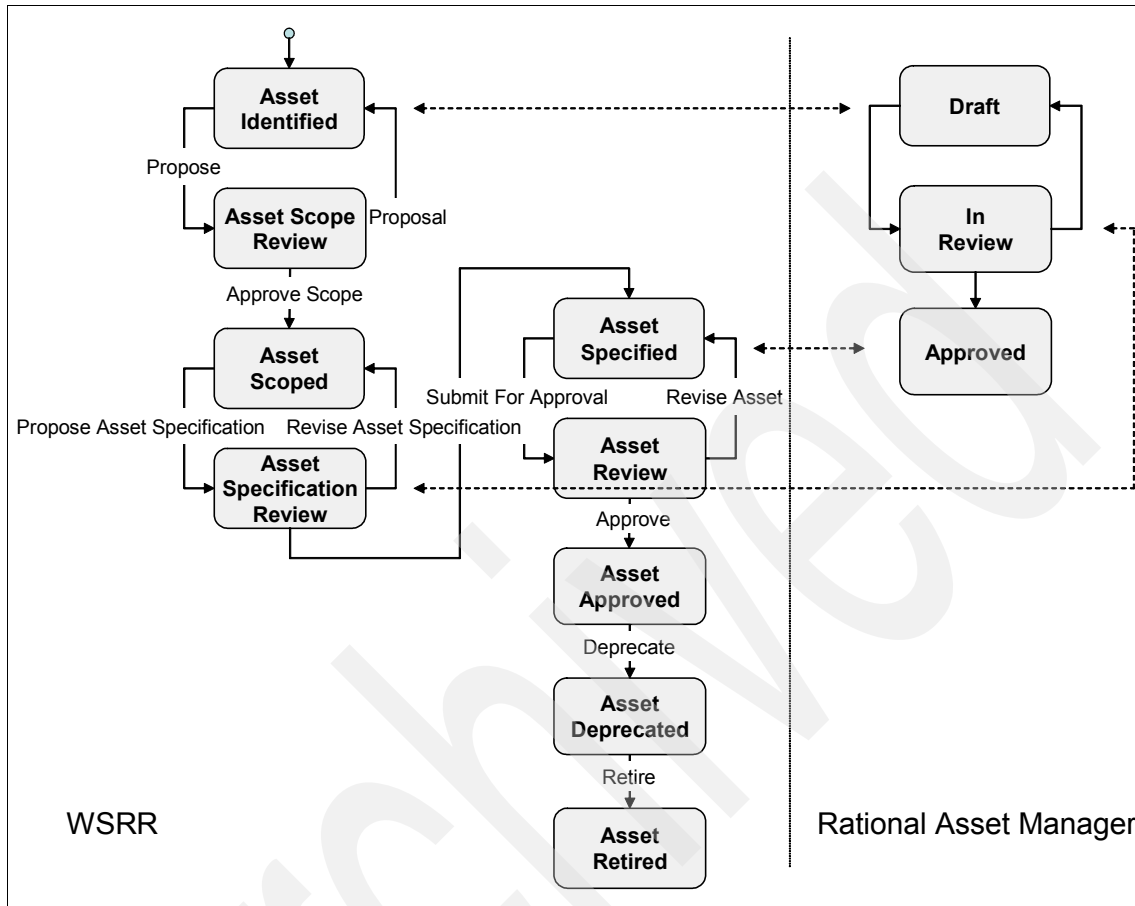


Figure 2-17 Asset lifecycle and synchronization with Rational Asset Manager

The recommended practice for objects in the Asset lifecycle is that those objects are first created in Rational Asset Manager. This order is important, because entities that are created in WSRR, and therefore automatically created in Rational Asset Manager during synchronization, can only be placed in one Rational Asset Manager community.

Further development of the asset and its artifacts can take place in WSRR and Rational Asset Manager environment, with changes reflected in WSRR Advanced Lifecycle Edition. The asset goes through the approval cycles in Rational Asset Manager, and artifacts can be attached and reflected across the systems. The lifecycle can continue in WSRR to further plan, approve, and deploy the service to production.

## 2.10 Additional WSRR-defined lifecycles

For reference and completeness, we have included lifecycle diagrams for the three other lifecycles that will be used throughout the scenarios in this book and that are relevant for the GEP entities.

### Endpoint lifecycle

This Endpoint lifecycle, which is shown in Figure 2-18, is used to govern a service endpoint to distinguish whether it is offline, online, or retired.

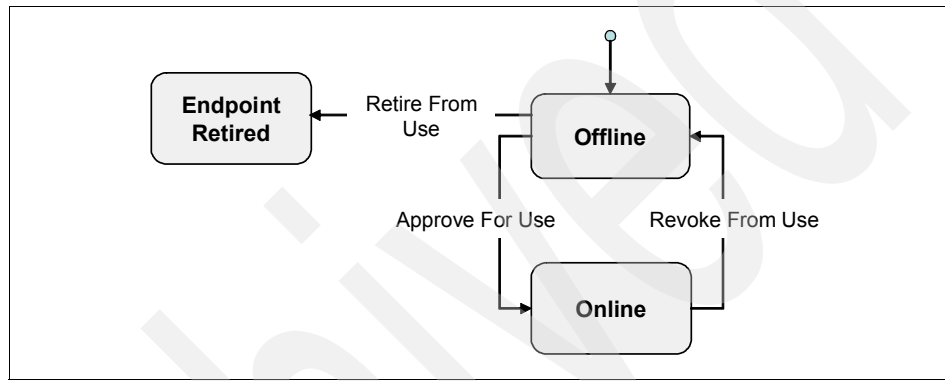


Figure 2-18 The Endpoint lifecycle

### Service Level Definition (SLD) lifecycle

This SLD lifecycle is used to govern a Service Level Definition from initial inception to its retirement.

Figure 2-19 on page 69 and Figure 2-20 on page 70 show the whole SLD lifecycle. Figure 2-19 on page 69 shows the model and assemble phases of the service lifecycle. Figure 2-20 on page 70 shows the deploy and manage phases.



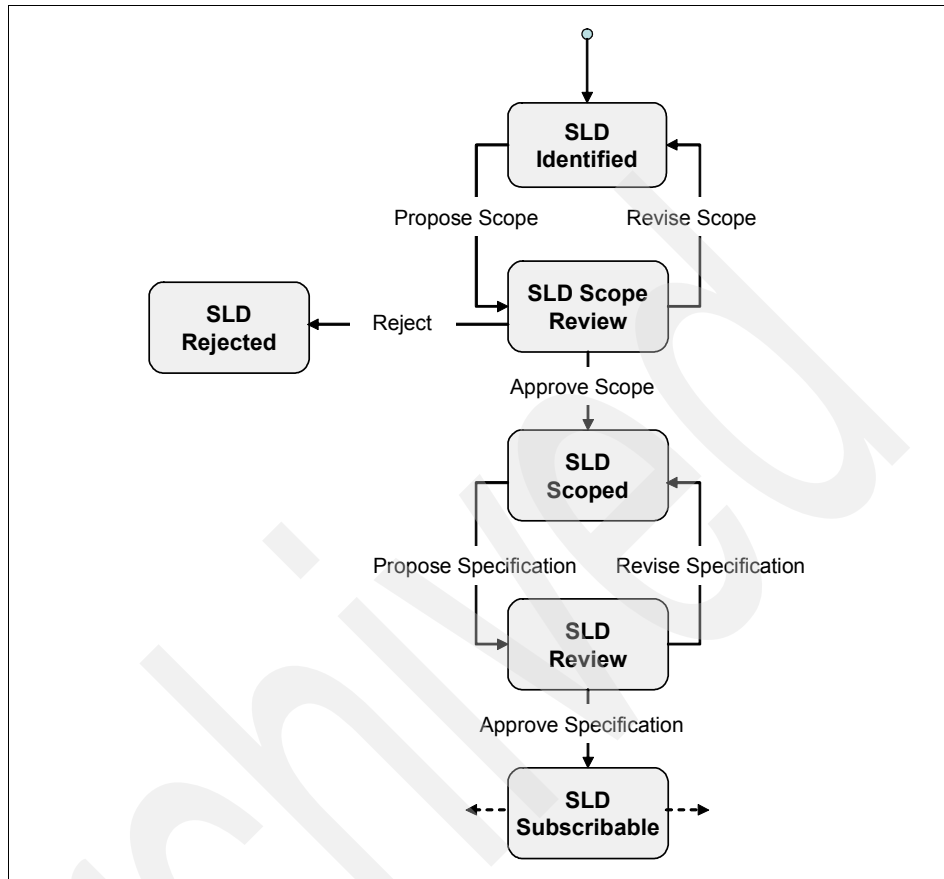


Figure 2-19 The SLD lifecycle: Model and assemble phases of the service lifecycle

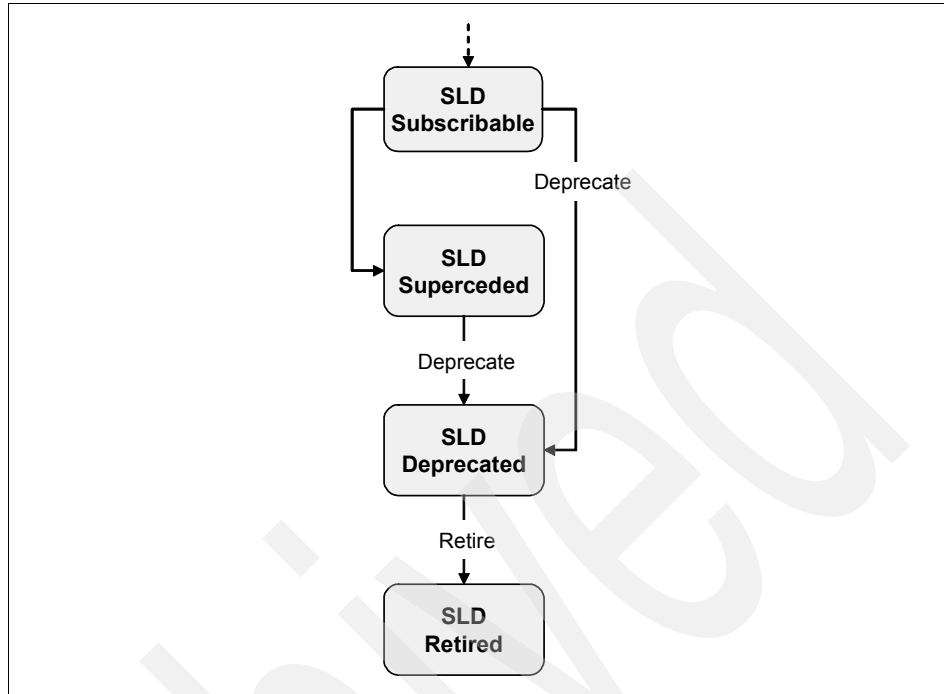


Figure 2-20 The SLD lifecycle: Deploy and manage phases of the service lifecycle

### Service Level Agreement (SLA) lifecycle

This SLA lifecycle, as shown in Figure 2-21 on page 71, is used to govern a service level agreement from initial inception to its rejection or termination. There is an active and inactive cycle, which allows it to be turned on or off. By making it active or inactive, a run time can determine whether a particular consumer is allowed to invoke a particular endpoint by checking whether the relevant SLA is in the correct state.

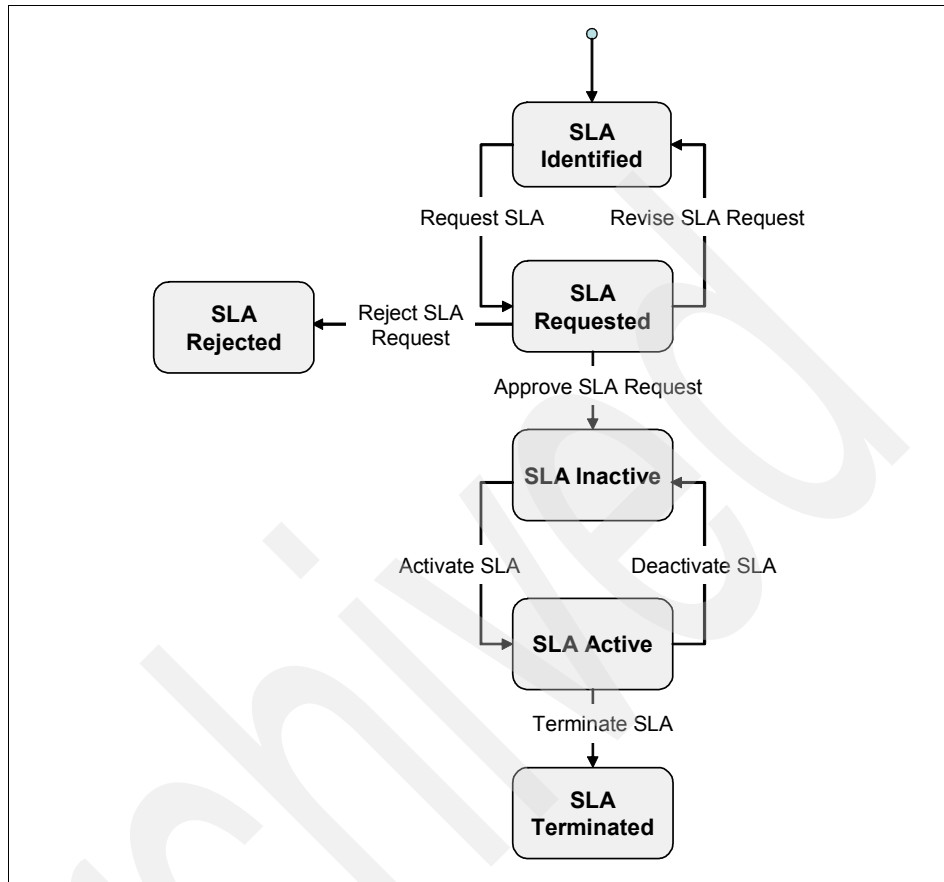


Figure 2-21 The SLA lifecycle

## 2.11 Additional resources

For more information, consult these resources:

- ▶ IBM WebSphere Service Registry and Repository Version 6.3 information center  
<http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp>
- ▶ IBM Rational Asset Manager Version 7.1.1.1 Information Center  
<http://publib.boulder.ibm.com/infocenter/ramhelp/v7r1m1/index.jsp>
- ▶ *Service Lifecycle Governance using WebSphere Service Registry and Repository V6.3*, SG24-7793

- ▶ *WebSphere Service Registry and Repository V6.3 Service Metadata Management and Governance*, REDP-4556
- ▶ *Integrating WebSphere Service Registry and Repository with WebSphere Process Server and WebSphere ESB*, REDP-4557
- ▶ *Integrating WebSphere Service Registry and Repository with WebSphere Message Broker and WebSphere MQ*, REDP-4558
- ▶ *Integrating WebSphere Service Registry and Repository with Tivoli® Composite Application Manager*, REDP-4560
- ▶ *Integrating WebSphere Service Registry and Repository with Tivoli Security Policy Manager*, REDP-4561
- ▶ *IBM Rational Software Rational Asset Manager v7.0 Capacity Planning and Configuration Guide*:

[http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/RAM\\_plan\\_config\\_guide\\_v4.0.5.pdf](http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/RAM_plan_config_guide_v4.0.5.pdf)

## Scenario overview

In this chapter, we give you an overview of several of the common scenarios within a service lifecycle and describe how these scenarios can be realized using WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition.

This chapter is divided into the following topics:

- ▶ 3.1, “Introduction” on page 74
- ▶ 3.2, “Top-down development” on page 74
- ▶ 3.3, “Bottom-up development” on page 75
- ▶ 3.4, “Service version management” on page 76
- ▶ 3.5, “Endpoint management” on page 77
- ▶ 3.6, “Reporting” on page 79

## 3.1 Introduction

In the previous chapters, we introduced the business and IT issues that drive the need for SOA governance, and we discussed the WSRR Advanced Lifecycle Edition product architecture and features that provide capabilities to address these needs. Before we can explore how these capabilities address the specific business and IT requirements and pain points, we must first elaborate on several of the specific business and IT use cases.

This chapter provides a high-level description of several of the primary use cases, which organizations execute in order to deliver the business value of a service-oriented architecture (SOA). We expect that a single organization will execute several, if not all, of these use cases at various times within their SOA adoption and the service lifecycle. We then describe each of these use cases in more detail in Part 2, “Top-down scenario” on page 95 and Part 3, “Other scenarios” on page 273, which provide specific examples of the steps that the roles within an organization undertake to execute that use case.

## 3.2 Top-down development

The starting premise of a top-down approach to development is that the business is driving the change. The organization requires a new feature or capability, and they need to initiate a process to develop this capability. In this scenario, the business users create a new representation of the required capability in WSRR. The creation of this representation triggers the Governance Enablement Profile-defined framework. This framework drives the development of the capability from business requirement definition, using a service governance lifecycle, through to the deployment of the capability in production.

This representation of business requirements in WSRR allows the governance process to directly associate the IT infrastructure with the business capabilities that it delivers. By utilizing this master metadata repository of SOA entities, the various roles can leverage the search capabilities to locate existing assets, which might be candidates for reuse.

The top-down approach to development is the primary focus of this book. We discuss the top-down approach to development in detail in Part 2, “Top-down scenario” on page 95.

### 3.3 Bottom-up development

*Bottom-up development* is the scenario in which an organization has existing services, applications, and processes defined within its IT infrastructure that need to be documented in WSRR Advanced Lifecycle Edition.

In this scenario, we use WSRR Advanced Lifecycle Edition to catalog the existing IT infrastructure. This approach to service definition in WSRR Advanced Lifecycle Edition is driven by the IT organization to expose the services that are available to the appropriate business users within the enterprise.

The IT organization instigates either a manual or automatic loading of the service definitions into WSRR. Then, the various roles undertake the steps necessary to ensure that WSRR Advanced Lifecycle Edition content is representative of the IT infrastructure in terms of the technical definition and the associated service metadata. The users ensure that the services are all in the stage of the lifecycle appropriate to where they are physically deployed within the organization. Additionally, the users leverage the capabilities of the governance enablement profile to ensure that the governance processes are enforced, even for services that were defined before these processes were implemented.

The most common usage of this development approach is when WSRR Advanced Lifecycle Edition is first deployed within an organization. At this time, the organization wants to catalog its existing SOA infrastructure in order to start enforcing governance within it. This process can either be achieved manually, with users loading and classifying artifacts individually, or by utilizing the Service Discovery capability of WSRR to automatically load and govern the existing, deployed service assets.

WSRR Advanced Lifecycle Edition automatically places the discovered artifacts into the appropriate lifecycle. Additionally, you can configure WSRR Advanced Lifecycle Edition to automate further steps as required, such as providing default metadata and performing lifecycle transitions. This flexibility allows existing IT environments to be cataloged quickly and governed, thereby delivering a rapid time to value.

## 3.4 Service version management

Service versioning is an important component of the management of service lifecycles. There are two types of service version modification, the naming of which varies between organizations but the definitions are common:

<b>Compatible</b>	A new version of a capability is required to deliver either enhanced functionality or to modify the underlying implementation. This new version is backward-compatible so that you can easily migrate existing service consumers to utilize the new implementation. This type of service version modification is often referred to as a <i>minor</i> version modification, for example, moving from Version 1.4 to Version 1.5.
<b>Incompatible</b>	A new version of a capability is required to deliver enhanced functionality, and this modification is not transparent to existing service consumers. This type of service version management is often driven by a need to change the interface that consumers use in order to invoke the capability. This type of service version modification is often referred to as a <i>major</i> version modification, for example, moving from Version 1.5 to Version 2.0.

### 3.4.1 Compatible version management

WSRR Advanced Lifecycle Edition provides the capability for a user to create a representation of a new version of a service. This representation allows the appropriate governance processes to be enforced and ensures that the user provides the information that is required by the service governance process.

A new version of an existing service, process, or application needs to be created when the business, development, or operations organization identifies a new feature as required, either as a technical modification or as a minor business requirement enhancement to an existing capability. This need for a new feature might be triggered by a variety of circumstances, for example:

- ▶ A new protocol needing to be supported
- ▶ A service defect requiring a change to the existing implementation
- ▶ A business unit requiring a minor capability to be added to an existing service, such as adding a verification method to a database creation service.



### 3.4.2 Incompatible version management

The development or operations organization might determine that a new implementation of a service is required, which does not provide a transparent migration path to existing service consumers.

In this situation, the user creates a new service version for an existing Business Capability and then uses a version number to appropriately identify the version as incompatible (for example, 2.x instead of 1.x).

### 3.4.3 Version numbering

An organizational decision must be made when defining internal service governance processes to determine the version numbering scheme to enforce. We recommend that you follow the version numbering format *<major>.<minor>.<service>*, where:

<b>major</b>	indicates an increment for incompatible changes
<b>minor</b>	indicates an increment for compatible changes
<b>service</b>	indicates an increment for program error fixes

## 3.5 Endpoint management

An important concept of an SOA implementation is the decoupling of the service consumers from the service providers. WSRR plays a pivotal role in this scenario by providing the master metadata repository that allows consumer services to locate the endpoint that they want to invoke. This decoupling mandates, therefore, that a process must be available to manage the modification of the provider endpoints, allowing for three main types of modification:

- |                            |   |
|----------------------------|---|
| <b>Endpoint relocation</b> | In this scenario, the service provider wants to change the physical location of the service. The service needs to be moved from one server to another server.   |
| <b>Retiring endpoints</b>  | In this scenario, an existing endpoint needs to be removed from service. This removal can be for a number of reasons, ranging from retiring a specific service version from use to removing a specific service instance to reduce the load on a given server. |

### **Additional endpoint deployment**

In this scenario, additional endpoints are required within a given environment. The two primary reasons for this type of deployment are either to improve the endpoint availability and performance characteristics or to provide an endpoint that supports additional capabilities, for example, additional communication protocols.

### **3.5.1 Endpoint relocation**

Endpoints within WSRR Advanced Lifecycle Edition have their own lifecycle, which means that they can be placed in and out of service very easily by the operations organization. When the new endpoint is deployed and verified, the operations team can quickly activate and deactivate the deployed endpoints as appropriate.

### **3.5.2 Retiring endpoints**

As service versions are superseded by new implementations, which deliver the same required Business Capability, the service governance lifecycle allows specific versions to be deprecated and, ultimately, retired. This process of removing services from the production environment is supported by the WSRR Advanced Lifecycle Edition lifecycle, which allows the operations organization to first deprecate existing implementations so that potential new consumers do not see the specific implementation. You can use the reporting and impact analysis capabilities in WSRR to allow the operations team to identify remaining service version consumers and ensure that they migrate onto the correct alternative version. After all of the consumers have been migrated, you can retire the service version from use and remove it from the deployment environment.

### **3.5.3 Deploying additional endpoints**

There are two primary reasons for the need to deploy additional endpoints:

- ▶ Endpoint availability and performance are supported by WSRR Advanced Lifecycle Edition by allowing the operations organization to deploy a new instance of a service to the appropriate environment and then manage the definition of that endpoint (and its availability to consumers) within WSRR. The most common need for this type of environment modification is to support additional service consumers (either reactively or proactively) based on service demand.

- ▶ Additional capabilities, which do not affect consumer/provider interoperability, are supported by the WSRR Advanced Lifecycle Edition-compatible version management that was discussed earlier in this chapter. For example, adding additional endpoints that support newly required communications protocols can be delivered using compatible version management.

## 3.6 Reporting

The monitoring of the service lifecycle is critical to organizations to measure the effectiveness of their governance processes. WSRR Advanced Lifecycle Edition allows you to define reports that can be used to identify various aspects of the solution effectiveness, for example:

- ▶ Identify the breakdown of business services by the owning organization.
- ▶ Determine the level of service reuse.
- ▶ Identify the number of service implementations in the various lifecycle stages.



## JKHL Enterprises scenario overview

This chapter introduces the case study that is used in the remainder of this book. This chapter provides an overview of the fictitious JKHL Enterprises company, its requirements, solution, organizational structure and roles, its success criteria, and how this project fits into the service governance solution.

We split these topics into the following sections:

- ▶ 4.1, “Introduction” on page 82
- ▶ 4.2, “An overview of JKHL Enterprises” on page 82
- ▶ 4.3, “Success criteria reporting” on page 91
- ▶ 4.4, “JKHL Enterprises project” on page 91

## 4.1 Introduction

Subsequent chapters of this book use a single example throughout the book to illustrate the use of WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition as the Service Governance solution.

JKHL Enterprises is a purely fictitious supply company that has specific goals and constraints that provide a typical set of challenges for reaching the full potential of service-oriented architecture (SOA) solutions. JKHL Enterprises is an example that appears in other IBM material. We have adapted the example in certain places to illustrate particular points that are relevant to building service governance with WSRR Advanced Lifecycle Edition.

## 4.2 An overview of JKHL Enterprises

JKHL Enterprises has approached IBM to assist in enabling an SOA governance solution. The company has adopted SOA to deliver a capability to rapidly respond to changing business needs by reconfiguring existing services and maintaining a clear mapping between business needs and IT implementations. As part of this SOA adoption, JKHL Enterprises thinks service governance is critical to success. JKHL Enterprises has decided that it needs a complete governance solution so that it can take full control of its services. JKHL Enterprises has already defined roles and communication plans and established an SOA Center of Excellence (CoE). Currently, JKHL Enterprises has a manual paper-based approach to service lifecycle governance and faces challenges that impede it from reaching its full potential.

JKHL Enterprises has identified the following pain points in its current solution that hinder it from reaching its goals:

- ▶ Inability to identify new service candidates and prioritize them
- ▶ Inability to analyze its SOA structure to create metrics, such as service reuse
- ▶ No well-defined process for the governance of changes and versions for the services
- ▶ Inability to tailor the view of the SOA assets based on role
- ▶ No integration between its development environment and runtime environment
- ▶ No integration with a code management repository

## 4.2.1 SOA governance vision and requirements

JKHL Enterprises (JKHLE) has captured a set of requirements for a service governance solution. JKHLE envisages a holistic closed-loop governance solution, which runs from design time through development time to run time.

### Design-time interaction

Currently, JKHLE has an improvised process for the introduction of new business requirements and for the design of the capabilities to fulfill those new requirements. Although JKHLE has IBM Rational RequisitePro® for the creation and tracking of requirements reported by its sales and marketing teams, there is no formal link between the requirements tool and the designs created. After the design of a capability is completed, the design documents are manually transferred to the development team.

Assessment of the potential opportunity for capability reuse cannot be performed in an optimal manner, because existing capabilities are not in a suitable state to be searched or queried. Additionally, formal requests to reuse services are often forgotten, and agreements on service reuse characteristics (such as the number of requests per hour) are not documented.

JKHLE sees the governance of design-time artifacts as critical to its SOA governance vision, because it acknowledges that increasing reuse will be a key factor in growing the profitability of the company. However, JKHLE thinks that this approach will only work if the reuse of capabilities is governed correctly; otherwise, it becomes extremely difficult to maintain control over the JKHLE capabilities.

### Development-time interaction

JKHLE uses the Rational product suite for the development of applications and services. Assets produced within the development process, such as code and service definitions in the form of Web Services Description Language (WSDL) files, are stored in a code repository.

JKHLE has no governance process for its development-time artifacts. Currently, the development manager verbally notifies the operations team that a new artifact is ready for deployment.

JKHLE thinks that its SOA governance solution needs to include governance at the development stage to facilitate the transitions from design, to development, to deployment.

## Runtime interaction

JKHLE uses the WebSphere Message Broker product as its Enterprise Service Bus (ESB).

Currently, the endpoints associated with a service are manually published to WebSphere Message Broker by the operations team. The endpoints are then manually updated when a new minor release of a service is developed, and new endpoints are manually published when a new major release of the service is developed.

JKHLE thinks that their SOA governance solution needs to allow dynamic endpoint selection based on version information to allow service requests to automatically use the most recent compatible release. JKHLE also thinks that the characteristics of its SOA, such as service reuse, need to be reportable by the registry and repository.

JKHLE plans to leverage the promotion capabilities of WSRR to ensure that the service definition artifacts are placed in the correct locations within its environment at the right points in the service lifecycle. You can obtain details about how to configure promotion in the WSRR Information Center at this Web site:

[http://publib.boulder.ibm.com/infocenter/sr/v6r3/topic/com.ibm.sr.doc/cwsr\\_promotion.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/topic/com.ibm.sr.doc/cwsr_promotion.html)

## Overall governance strategy

In addition to the requirements described at the capability lifecycle stages, JKHLE has further requirements that must be addressed to allow the company to achieve a holistic closed-loop governance solution:

- ▶ Ability to govern the service lifecycle from new service creation to service retirement using integrated tooling
- ▶ Have an integrated registry and repository that automatically federates the design-time and run-time artifacts as required
- ▶ Ability to provide a prescriptive framework for service versioning

### 4.2.2 SOA governance solution

JKHLE found that WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition meets its requirements. By implementing service governance with WSRR Advanced Lifecycle Edition, JKHLE will also gain value from these capabilities:

- ▶ Governing new service creation and determining the priorities and the funding for the establishment of these services



- ▶ Identifying who owns services to drive better requirements against services to foster trust and engender reuse
- ▶ Ensuring that a consistent service change management process reduces potentially critical impacts on its business
- ▶ Providing a versioning process for services that will reduce version changes for capability consumers and eliminate uncontrolled numbers of capabilities in production, thereby reducing operational and maintenance costs
- ▶ Managing service consumers, enabling more efficient capacity planning
- ▶ Increasing ESB flexibility
- ▶ Providing reports that can give management insight to the business value of its SOA and operational metrics

Figure 4-1 shows the complete SOA governance vision with the products that enable it.

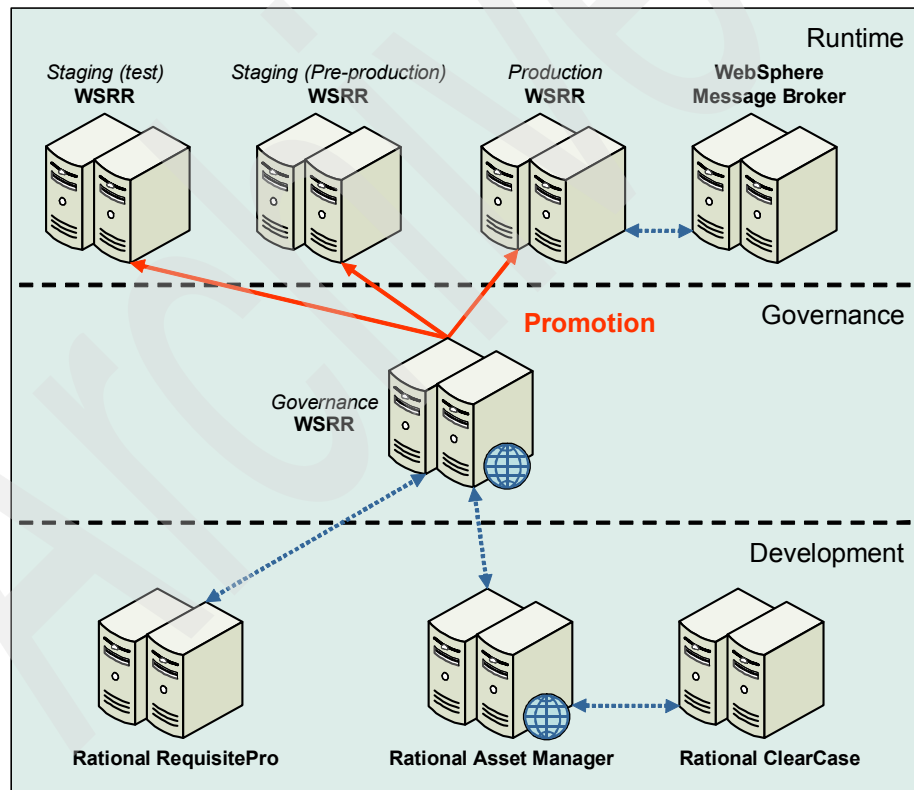


Figure 4-1 JKHLE Service Governance topology

### 4.2.3 Organizational structure

JKHLE has a simple organizational structure:




- ▶ JKHL Enterprises: A top-level organization, which represents the complete enterprise
- ▶ Shared services: A child organization of JKHLE, which represents the departmental team that is responsible for the development and delivery of services that are shared across the enterprise
- ▶ Sporting goods: A child organization of JKHLE, which represents the Line of Business (LoB) team for the Sporting Goods department
- ▶ Footwear: A child organization of JKHLE, which represents the LoB team for the Footwear department
- ▶ Clothing: A child organization of JKHLE, which represents the LoB team for the Clothing department

Distinct roles exist within the JKHLE organization. We use these roles throughout future chapters, so the full list is presented here for reference.

#### JKHLE cross-organizational roles

Table 4-1 shows the JKHLE cross-organizational roles.

























Table 4-1 Cross-organizational roles

Role	
Business Analyst for JKHLE	 Business Analyst
SOA Governance CoE chairman	 SOA Governance
Administrator responsible for configuring WSRR Advanced Lifecycle Edition	 Administrator

## Departmental roles

Table 4-2 shows the JKHLE departmental roles.

Table 4-2 Departmental roles

Role	Sporting goods	Shared services	Footwear	Clothing
Business unit leaders	 LoB Manager	 LoB Manager	 LoB Manager	 LoB Manager
Development team leaders	 Development Lead	 Development Lead	 Development Lead	 Development Lead
Release and project managers	 Release Manager	 Release Manager	 Release Manager	 Release Manager
Operations team leaders	 Operations	 Operations	 Operations	 Operations
Developers	 Developer	 Developer	 Developer	 Developer
Testers	 Tester	 Tester	 Tester	 Tester

## 4.2.4 Organizational structure representation in WSRR

To replicate the organizational structure of JKHLE, the WSRR Advanced Lifecycle Edition administrator creates Organization entities within WSRR. Figure 4-2 shows the administrator creating the Shared Services organization entity.

The screenshot displays the 'Service Registry and Repository' (WSRR) interface. The top navigation bar includes the JKHLE logo, the title 'Service Registry and Repository', and user information: 'Perspective: Administrator', 'richard', and a 'Logout' link. Below the navigation bar is a search bar and a 'Help' link. The main content area is titled 'Organization' and shows the breadcrumb 'Organization > Shared services > New Organization'. A message states: 'Create a new entity of type: Organization. When you have specified all required property values, and relationship targets, click 'Finish'.' Below this is a 'Finish' button and a 'Cancel' button. A table shows the 'Shared services' organization with a 'Targets' column and a 'Status' column. The 'Status' column shows a green checkmark and the word 'New'. Below the table is a link 'Add Relationship' and a link 'Add Organization'. A 'Child Organizations' section is also visible. The 'Details' section is expanded, showing 'General Properties' and 'Relationships'. The 'General Properties' section includes fields for 'Name' (Shared services), 'Description' (The team that develops and maintains the services that are reused across JKHL Enterprises), 'Contact' (nickie), and 'Contact Email' (nickie@jkhle-enterprises.com). The 'Relationships' section shows 'Child Organizations' as 'None'. A 'Field help' box on the right explains the 'Contact Email' property. A 'Page help' box at the bottom right provides more information about the page.

**Organization**

**Organization > Shared services > New Organization**

Create a new entity of type: Organization. When you have specified all required property values, and relationship targets, click 'Finish'.

[Finish](#) [Cancel](#)

Targets	Status
<a href="#">Shared services</a>   <a href="#">Add Relationship</a>	<input checked="" type="checkbox"/> New

[Child Organizations](#) [Add Organization](#)

**Details**

[Edit Classifications](#)

**General Properties**

**Name**  
Shared services

**Description**  
The team that develops and maintains the services that are reused across JKHL Enterprises

**Contact**  
nickie

**Contact Email**  
nickie@jkhle-enterprises.com

**Relationships**

**Child Organizations**  
None

**Field help**  
The Contact Email property identifies an email address used to contact or notify any requests for actions that affect assets or resources owned by this organization.

**Page help**  
[More information about this page](#) (Opens in a new window.)

Figure 4-2 WSRR Advanced Lifecycle Edition administrator creates the Shared services Organization entity

After the administrator creates the Organization entities within WSRR, the structure is complete, as shown in Figure 4-3 on page 89.

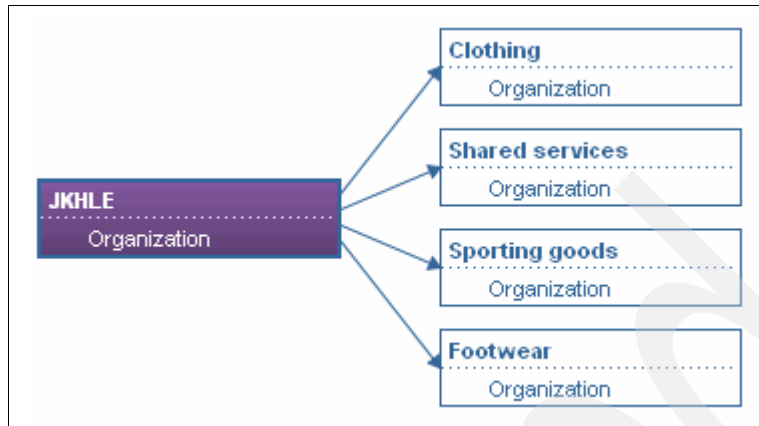


Figure 4-3 Organizational structure defined in WSRR

## 4.2.5 Organizational structure within Rational Asset Manager

The WSRR Advanced Lifecycle Edition Administrator creates four communities within Rational Asset Manager to represent the four development teams within JKHLE: the Shared services development team, the Sporting goods development team, the Footwear development team, and the Clothing development team. The administrator then supplies access to the communities for the correct members of JKHLE.

All members of the community are able to produce and consume assets within Rational Asset Manager. Additionally, the business users are able to administer forums.

## 4.2.6 Customizations

JKHLE has made several customizations to its WSRR and Rational Asset Manager.

### Themes

JKHLE prefers a common look between the WSRR and Rational Asset Manager tools. The company created a common look by creating a theme for both the WSRR and Rational Asset Manager tools. You can obtain more information about themes at “Theme customization” on page 360.

### Charter plug-in

Charter documents are required to be attached to Business Capabilities in WSRR as the target of two relationships: the Charter relationship and the

Artifacts relationship. The latter is needed to allow the asset to be correctly synchronized between WSRR and Rational Asset Manager. A plug-in has been developed that will automatically add any target of the Charter relationship to the Artifacts relationship. You can obtain further details about the plug-in in “Charter validator” on page 361.

## 4.2.7 Governance review process

There are four types of assets that are reviewed within Rational Asset Manager: Business Solution, Release, Specification, and Document of Understanding. JKHLE has decided that it requires a tailored review process for each of the asset types. Additionally, it is possible to differentiate review processes at the categorization level<sup>3</sup>, but JKHLE does not feel that its review processes require this level of differentiation.

JKHLE has decided to have a review board-based approach to reviewing assets. The review board is a user or a group of users who are able to further tailor the review process to include additional reviewers on an asset by asset basis. After all of the reviews have been collated, the review board is able to make the final decision on whether to approve or reject the reviewed asset. JKHLE has also specified reviewers who need to be involved in the review process for all assets of a certain type.

The basic structure for their review processes is set this way:

- ▶ Business solution:
  - Review board: SOA Governance CoE
  - Specified reviewers: Business unit leader for community
- ▶ Specification
  - Review board: SOA Governance CoE
- ▶ Release:
  - Review board: Business unit leader for community
  - Specified reviewers: Release manager for community
- ▶ Document of Understanding:
  - Review board: Business unit leader for community
  - Specified reviewers: Release manager for community

<sup>3</sup> Several of the asset types in Rational Asset Manager are subdivided using categorization. It is possible to use this categorization in Rational Asset Manager to control the selected review process for the asset. For example, Business Solutions categorized as Applications can follow a review process that differs from the review process for Business Solutions categorized as Service.

## 4.3 Success criteria reporting

JKHLE wants to measure the success of its service governance solution so that the company can make informed decisions around the future use of the solution and adaptations that it might choose to make.

JKHLE has chosen to report on the usage of the service governance solution using three metrics:

- ▶ Service reuse: A report, broken down by Business Capability, that shows how many services the Business Capability reuses. An increase in the amount of reuse over time shows that the service governance solution is reducing the cost of service duplication.
- ▶ Business capability by lifecycle: A report showing how many business capabilities are in each lifecycle state. This report allows JKHLE to see how many capabilities are governed by its service governance solution.
- ▶ Business capability by organization: A report showing how many business capabilities have been created and approved by each organization within JKHLE. This report allows JKHLE to monitor the uptake of the service governance solution by the various organizations within the company.

## 4.4 JKHL Enterprises project

JKHL Enterprises has a key project that it thinks is a good opportunity to fully exercise the service governance solution that it has put in place using a top-down approach. The project involves significant integration between both existing and newly required services, processes, and applications. Therefore, the control of capabilities that the service lifecycle governance offers will ensure that the complexity of the project does not become an issue.

### 4.4.1 JKHL Enterprises sports equipment Web site

Currently, JKHLE has a Web site on the Internet that catalogs a range of top-end sports equipment that is available for purchase. Customers of the site are able to populate a virtual shopping cart with the products that they want to purchase. When the customers complete their shopping, they proceed to a virtual checkout where they fill out their payment details, delivery address, and e-mail address. Finally, the customer is invited to submit their details, and they are presented with a window specifying that their order is being processed and that they will be contacted upon completion.

At this point, unknown to the customer, an e-mail is sent to the sales manager of the sports equipment department. The manager coordinates the customer status check, phones the bank to take the customer payment, accesses a shipping company's Web site to organize shipment, and finally sends an e-mail to the customer to let them know of the successful completion of their order.

#### **4.4.2 JKHL Enterprises sports equipment Web site upgrade**

The sports equipment Web site has been identified by the upper management in JKHLE as an opportunity to save costs. They also see sports equipment sales as a real opportunity for growth and have these concerns:

- ▶ The use of a manual ordering process will limit scalability.
- ▶ Customers will be disappointed that the order confirmation (or rejection) is not reported to them immediately.
- ▶ At peak times of the year, the order delay can be several days, which impacts the shipping time to customers.

The identified solution is to create an integrated order processing business application that makes calls to three services to process the order. The three services already exist within the JKHLE service governance solution, after being added using the bottom-up approach, but currently are invoked manually by the Sporting Goods sales manager.

##### **Customer Status Check service**

The Customer Status Check service looks at the customer's history with JKHLE and reports on any issues that might be of concern. The service has been fully designed and developed and is already in production.

The Ordering application will be able to make use of the existing service and will need to put service level agreements (SLAs) in place to ensure that the quality of service that is required by the Ordering application is satisfied.

##### **Execute Payment service**

The Execute Payment service withdraws the necessary funds from the supplied account and reports on any issues that might be of concern. The service has been fully designed and developed and is already in production.

The Ordering application will be able to make use of the existing Execute Payment service and will need to put SLAs in place to ensure that the quality of service required by the Ordering application is satisfied.



### **Ship Goods service**

The Ship Goods service is also a service that already exists within the JKHLE organization. Currently, it automates the shipping of goods to any location within the U.S.

The Ordering application will be able to make use of the existing Ship goods service; however, additional requirements on the service might be identified during the modelling phase of the Ordering application development.

Archived





## Part 2

# Top-down scenario

In the next chapters, we look in detail at the various stages in implementing a top-down development approach while leveraging the WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition capabilities.

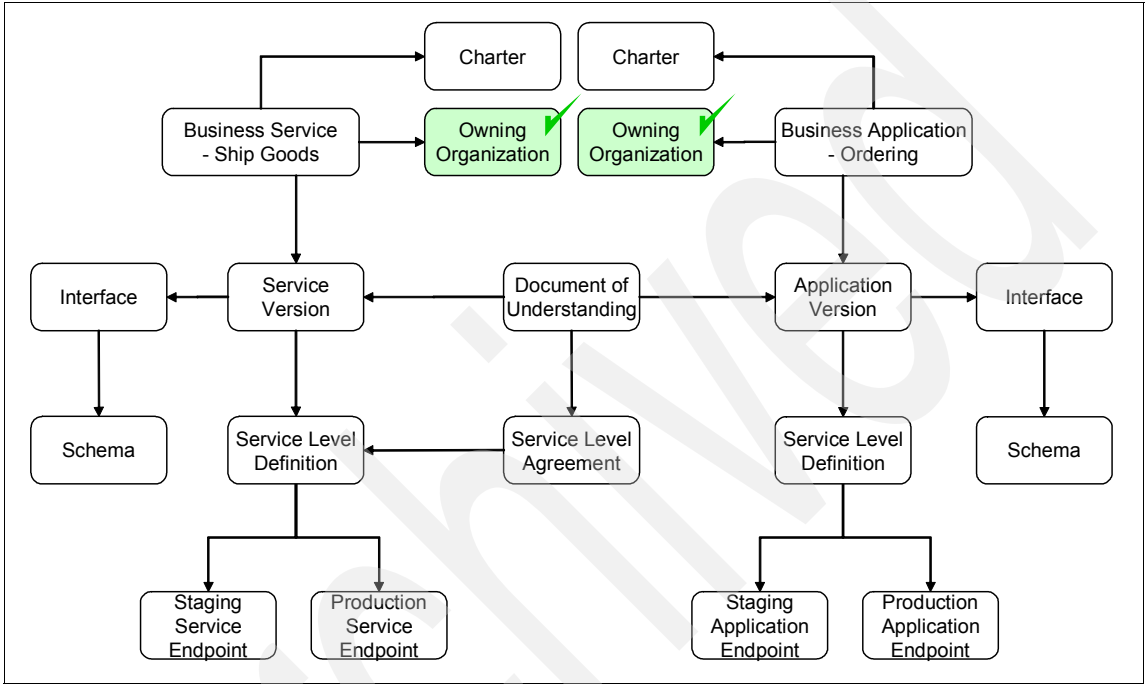
In Chapter 3, “Scenario overview” on page 73, we introduced the basic concepts of the top-down development approach by taking a deeper look at the steps involved in implementing this approach while utilizing the capabilities of the WSRR Advanced Lifecycle Edition solution. At each stage in this development approach, Chapter 3, “Scenario overview” on page 73 indicated the user role that we propose will execute the steps in the process.

The chapters in this part describe the top-down approach for our scenario:

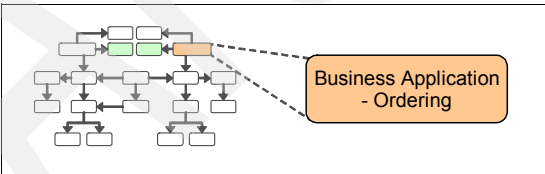
- ▶ Chapter 5, “Defining the business requirements” on page 99
- ▶ Chapter 6, “Elaboration and review of business requirements” on page 109
- ▶ Chapter 7, “Release specification” on page 129
- ▶ Chapter 8, “Defining Service Level Agreements” on page 185
- ▶ Chapter 9, “Release development” on page 217
- ▶ Chapter 10, “Application staging” on page 239
- ▶ Chapter 11, “Production deployment” on page 255

Progress key guide

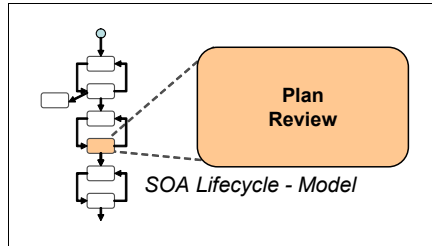
In this part of the book, we show the progress in building the entities in the governance model by using a depiction of the model. This chart shows the components of the Governance Enablement Profile (GEP) business model, which was introduced in 2.7, “Service governance assets” on page 48, that will be created as we progress through the scenario.



We also show the entity that is going to be developed in the current step of the process.



We similarly display the lifecycle state of the entity under development in the current step of the process.



Archived

# Defining the business requirements

This chapter describes the steps that the business users execute to define the business requirements for the Ordering application. This chapter includes these topics:

- ▶ 5.1, “Introduction” on page 100
- ▶ 5.2, “Business value definition” on page 100

## 5.1 Introduction

As described in 4.4, “JKHL Enterprises project” on page 91, JKHL Enterprises (JKHLE) has identified the Sports equipment Web site as the first project that will use the top-down approach leveraging the WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition solution. The first step in employing this approach is to clearly define the business requirements for the Web site enhancements.

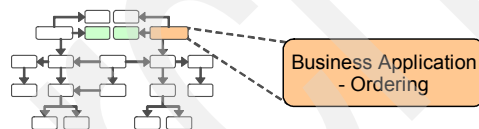
## 5.2 Business value definition



A business analyst in the Sporting Goods Line of Business (LoB), in conjunction with the LoB manager, develops the business case for the proposed project to explain the value proposition. This business case details the business pain points that the new Ordering application must be designed to overcome, which were detailed in 4.4, “JKHL Enterprises project” on page 91.

The analyst creates a new requirement in JKHLE’s Rational RequisitePro, the existing requirement management system, to externalize the requirement and the associated business drivers within the JKHLE organization.

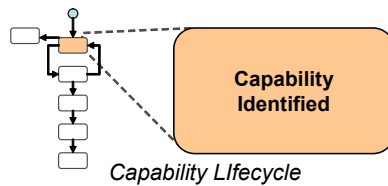
### 5.2.1 Business Capability definition



After defining the business case and the business-level requirements for the new version of the Ordering application, the business analyst now needs to instigate the review of this

proposal. Before taking this step, the business analyst first needs to create a representation of the new capability version of the Sporting Goods Ordering application in Rational Asset Manager. This representation is used to collate the various requirements, business case, and business charter definitions that are ready for review by the business and governance stakeholders. It represents the Business Capability that the application fulfills, rather than a particular implementation of the application.





When it is created in Rational Asset Manager, the representation is automatically synchronized into WSRR and placed into the Business Capability lifecycle. This lifecycle starts to enforce governance and to provide a framework of actions, which the user can follow to guide the service through the service governance lifecycle.

The business analyst creates a Business Solution in Rational Asset Manager as shown in Figure 5-1 to define the basic specification:

<b>Name</b>	Ordering Application
<b>Version</b>	2
<b>Community</b>	Sports department
<b>Asset type</b>	Business Solution
<b>Short description</b>	Describes the capability that the new application will provide

Figure 5-1 Business analyst creates a Business Solution in Rational Asset Manager

The recommended practice for all entities that will be synchronized between WSRR and Rational Asset Manager is that those objects are first created in Rational Asset Manager. This is because it is only possible to configure the synchronization to replicate assets created in WSRR to a single Rational Asset Manager community. Therefore, in order to overcome this limitation, the entities

must always be created in Rational Asset Manager first to ensure that they are associated with the correct community.

After documenting the requirements in the existing requirement management tool, the business analyst now adds a link to these requirements in the new Ordering application. The requirements are linked to the Business Solution as shown in Figure 5-2.

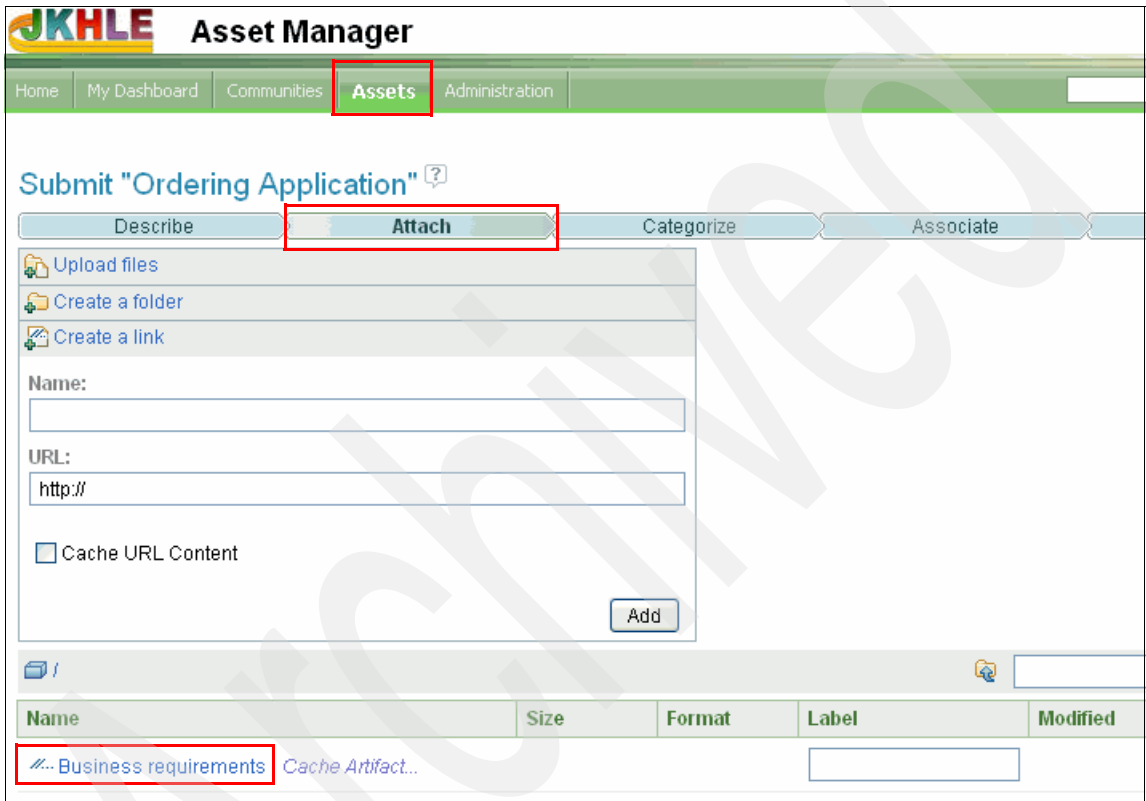


Figure 5-2 Business analyst creates a link to the identified requirements

Finally, the analyst categorizes the Business Solution as a Web Application (as shown in Figure 5-3 on page 103). This step identifies the Business Solution as a Business Application, and the corresponding object type of Business Application will be created in WSRR.

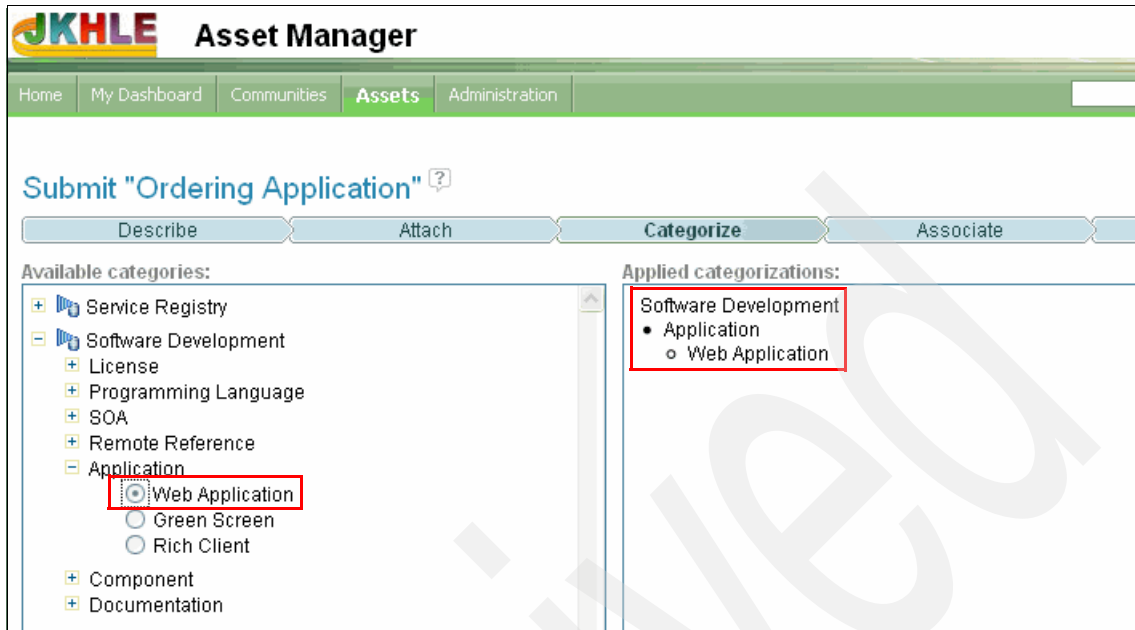


Figure 5-3 Business analyst adds the Web Application categorization

The asset is stored using “Save as Draft,” which is required because work needs to be done in WSRR before the Business Solution can be proposed for review.

**Tip:** Always use the “Save as Draft” option when creating new assets in Rational Asset Manager.

**Note:** When the Business Solution is created in Rational Asset Manager, a corresponding Business Capability is automatically created in WSRR, and the two entities are linked together. In this case, the Business Solution was categorized as a Web Application. Therefore, a specialization of Business Capability, in this case a Business Application, is created in WSRR.

For more information about entity synchronization and classification, see 2.8, “WSRR Advanced Lifecycle Edition synchronizable assets” on page 52.

The business analyst then uses service decomposition techniques to understand the basic structure of the existing process and identify the three specific services that the process will need to utilize. It is determined that the integrated ordering process needs to provide the following functional capabilities:


- Check that the customer has a valid JKHL Enterprises account.

- ▶ Take payment from the customer and verify that this payment completes successfully.
- ▶ Send the order to the warehouse for processing and shipment to the customer.



The business analyst searches the Rational Asset Manager for other business processes, applications, and services, which might be impacted by the creation of this new process. Three existing services are discovered that can be used to automate the Ordering application:

- ▶ Customer Status Check: Verifies that the customer has an active account with JKHL Enterprises
- ▶ Execute Payment: Executes the transaction with the appropriate payment provider
- ▶ Ship Goods: Sends the order to the warehouse for processing and shipment to the customer

The business analyst creates a new forum within Rational Asset Manager and links it to the three identified services and the newly created Business Solution, with a topic created to discuss the intention to consume the three identified services. This step is shown in Figure 5-4 on page 105.


**Asset Manager**

[Home](#)
[My Dashboard](#)
[Communities](#)
[Assets](#)
[Administration](#)

Asset: **Ordering Application [2]**



[General Details](#)
[Content](#)
[Ratings](#)
[Forums](#)
[Statistics](#)
[Download](#)

Forums >

**Forum: Discuss new sports ordering application**

[Change Management Configuration](#) ?

Configuration: none
[Configure...](#)

[Assets Using This Forum](#) ?

Asset Name	Version	Type	Community	Modified
<a href="#">Execute payment</a>	1	Business Solution	Shared services	Aug 3, 2009
<a href="#">Customer status check</a>	1	Business Solution	Shared services	Aug 3, 2009
<a href="#">Ship goods</a>	1	Business Solution	Shared services	Aug 3, 2009
<a href="#">Ordering Application</a>	2	Business Solution	Sports department	Aug 3, 2009

[Link Asset...](#)

[Topic List](#) ?

Title	Created by	Replies	Updated ↓	
<a href="#">Dependent services</a>	<a href="#">bob</a>	0	Aug 3, 2009 4:42 PM	<a href="#">Delete</a>

[New Topic...](#)

Figure 5-4 Business analyst creates a forum to discuss the identified reuse

The business analyst also posts a message on the forum to briefly describe the proposed enhancements to the Ordering application as shown in Figure 5-5 on page 106.

The screenshot shows the JKHLE Asset Manager web application. The top navigation bar includes links for Home, My Dashboard, Communities, Assets (highlighted), and Administration. Below this, the asset name 'Asset: Ordering Application [2]' is displayed with edit and delete icons. A secondary navigation bar contains tabs for General Details, Content, Ratings, Forums (selected), and Statistics, along with a Download button. The main content area is titled 'Create Topic' and contains a form with the following fields:

- Title:** A text input field containing 'Sports equipment Web site enhancements'.
- Description:** A rich text editor with a toolbar (including icons for undo, redo, bold, italic, underline, bulleted list, numbered list, link, unlink, and expand) and a text area containing:

We plan to automate the customer ordering Web site so that customer account checking, payment and shipping are all automated.  
This is being done to tackle the scalability issues, improve customer satisfaction and reduce processing time.  
Thoughts and opinions welcome
- Attachment:** A text input field with a 'Browse...' button next to it.

At the bottom right of the form are 'Create' and 'Cancel' buttons.

Figure 5-5 Business analyst adds a post discussing the proposed enhancements

## Define nonfunctional requirements

The business analyst also details the basic nonfunctional requirements that the new ordering service must fulfill:

- ▶ **Scalability:** The process must be scalable to support the anticipated growth in customer demand.
- ▶ **Customer Satisfaction:** The project needs to increase satisfaction ratings.
- ▶ **Processing time:** The project must reduce the maximum processing time from several days to 4 hours.

These requirements are driven by the business case that was initially created. The business analyst records these requirements in the requirements tool that was used previously.

## 5.3 Progress summary

As shown in Figure 5-1 on page 101, we have started the definition of the Ordering Business Application. As described in 4.2.3, “Organizational structure” on page 86, the organization hierarchy has already been defined in WSRR, so the business analyst is able to identify the owning organization of the new application. In this case, the Sports department owns the new application. The business analyst creates a relationship between the owning organization and the new application in WSRR.

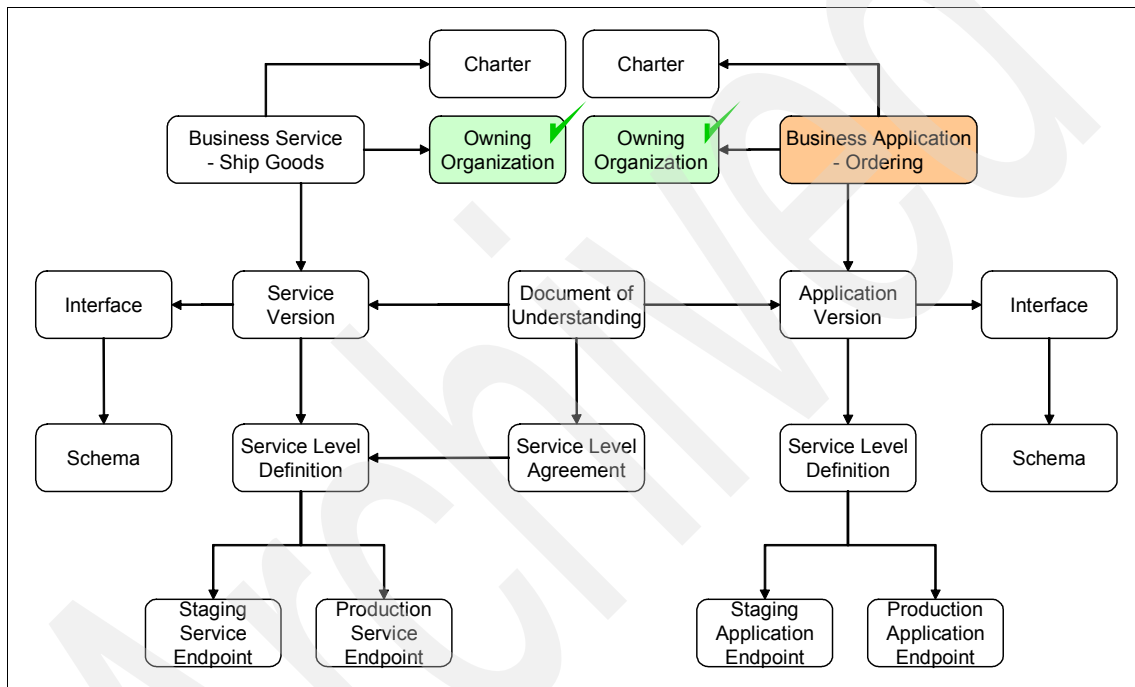


Figure 5-6 New Business Ordering Application has been created in this step of the process





## Elaboration and review of business requirements

This chapter looks at how the business analyst uses the WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition to extend the business requirements and instigate the development processes.

This chapter includes these topics:

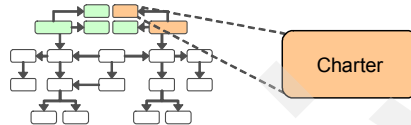
- ▶ 6.1, “Introduction” on page 110
- ▶ 6.2, “Business application charter definition” on page 110
- ▶ 6.3, “Progress summary” on page 119
- ▶ 6.4, “Application scoping and investigation” on page 120
- ▶ 6.5, “Creating Documents of Understanding” on page 125
- ▶ 6.6, “Progress summary” on page 126

## 6.1 Introduction



In Chapter 5, “Defining the business requirements” on page 99, the business analyst created a Business Solution entity in Rational Asset Manager to represent the new Ordering application. The business analyst now needs to select an owning organization for the application and to develop a charter document detailing the use case for the application. Figure 6-1 on page 111 shows the related Governance Enablement Profile (GEP) artifacts and that the Business Capability is still a work in progress.

## 6.2 Business application charter definition



After creating an entity to represent the Business Capability and providing a basic description of the new application, the business analyst now needs to elaborate this definition by creating a *charter*<sup>4</sup>, as

shown in Figure 6-1 on page 111. This charter describes the service capabilities and ownership, including an overview of all functional and nonfunctional requirements. Note that these requirements apply to all versions of this capability. Any detailed requirements for a particular version are placed on the specific capability version, which we will see with the required enhancements to the Ship Goods service.

The business analyst creates the charter as a text document, containing a detailed description for each of the following service requirements:

- ▶ Account verification: Ensure that the customer has an active account.
- ▶ Payment processing: Verify payment details and execute the transaction with the relevant payment provider.
- ▶ Payment verification: Verify that this payment completed successfully, and store the payment confirmation number in the customer order record database.
- ▶ Shipment processing: Send the order to the warehouse for processing and deliver to the customer.
- ▶ Scalability: The application must support deployment within a clustered environment.

<sup>4</sup> A charter is created by the business to describe the functional and nonfunctional requirements that are driving the need for a new Business Capability.

- ▶ Performance: The payment verification step of the process must complete within 30 seconds.
- ▶ Transaction feedback: The ordering process must give feedback about the status of the customer order immediately after payment verification has completed.
- ▶ Performance: Each instance of a customer order must complete within 4 hours.
- ▶ Ownership: Define the organizational ownership for the application, including proposed individual owners for the technical and business aspects of the service.

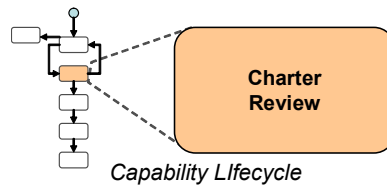
The business analyst then attaches this document to the Ordering application definition in WSRR, as shown in Figure 6-1.



Figure 6-1 Business analyst attaches the charter document to the Ordering application

**Note:** When the charter document is added to the Charter relationship, it is also automatically added as a target of the Artifacts relationship. All targets of the Artifacts relationship are automatically synchronized with Rational Asset Manager. In this instance, it is preferable that the charter document is available in both WSRR and Rational Asset Manager. JKHLE implemented this customization during its installation of the WSRR Advanced Lifecycle Edition solution.

**Note:** You might think that it is preferable to attach the charter document to the entity in Rational Asset Manager rather than to the entity in WSRR. However, due to the design of the synchronization capabilities, it is only possible to automatically synchronize the reference to the charter document from both WSRR and Rational Asset Manager entities if the charter is attached to the WSRR entity.



After loading the charter into the registry and assigning an owning organization to the Business Application, the analyst has completed the initial definition of the application. It is ready for review by the service-oriented architecture (SOA) Center of Excellence (CoE) board. In order to indicate

its readiness for review and to make it available for the reviewer's actions, the analyst must transition the business application to the next state, Charter Review, in the Business Capability lifecycle in WSRR. Figure 6-2 on page 113 shows the detailed view (Details tab) of the Ordering application in WSRR; the highlighted button (Propose for Charter Review) is used by the business analyst to transition the application into the next state.

JKHLE

Service Registry and Repository

Perspective: Business

Home

Actions

View

Tasks

My Service Registry

Help

Business Application

Business Applications > Ordering Application

Detail view for Business Application. A Business Application represents a business capability that is realized through a legacy IT applications OR a particular channel to market realized by a Web or Portal Application.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering Application

Description

Automate customer ordering application on the sports equipment Web site

Business Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B6190A866-F3B3-7ECD-9BD4-FF7D3A981C9E%7D&w=2

Remote State

Draft

Owner Email

Additional Properties

Back

Propose for Charter Review

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Charter

Ordering application charter.doc

Service Interface Versions

None

Versions

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

Ordering application charter.doc

Dependent Entities

Dependent Assets

Figure 6-2 Business analyst proposes the Ordering application for charter review

At this point in the service governance lifecycle, the business analyst has created the Business Application and Charter entities in WSRR, as shown in Figure 6-3 on page 114.

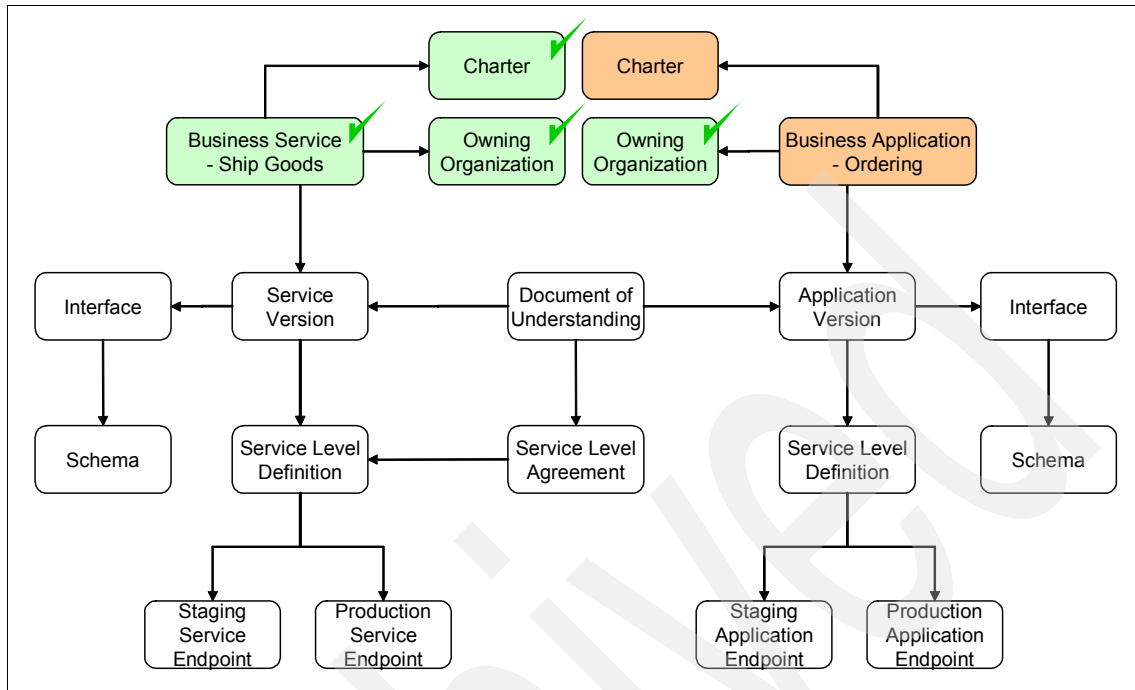


Figure 6-3 Current state of the GEP modeled entities for the Ordering application

## 6.2.1 Business Application charter review



The SOA Governance CoE is notified of the Business Application charter being proposed and, as the review board member, is able to tailor the reviewers of the asset. By default, the Sporting Goods LoB manager is the only reviewer. When the reviewers have been defined, the SOA Governance CoE representative starts the Rational Asset Manager review process.

The Sporting Goods LoB manager is notified of the Business Application charter proposed by the business analyst. Using the Rational Asset Manager, the manager reviews the business requirements and the charter that have been proposed.

Rational Asset Manager provides the option to Reject or Approve the definitions. If the sponsor determines that the charter or requirements require further elaboration, the sponsor can provide a comment describing the required updates and return the asset to the business analyst for additional work, as shown in Figure 6-4 on page 115. If the review board agrees with the decision to reject the proposal, the business application is returned to the draft state and the business analyst reviews the suggested modifications, performing the steps in 6.2,

“Business application charter definition” on page 110 again to attach the new version of the charter.

The Business Application definition in WSRR only shows the most recent charter that is attached to it. However, the Rational Asset Manager maintains a record of all versions of the charter definition so that reviewers can see the entire history of changes to the Business Solution definition.

**JKHLE Asset Manager**

Home My Dashboard Communities **Assets** Administration

Asset: **Ordering Application [2]** ?

General Details Content Ratings Forums **Review** Statistics [Download](#)

**Solution review 8/3/09** [Review history](#)

Complete the review and click Accept or Reject. [Review](#) [Actions](#) ▼

► **Policy Compliance Results :** (1 policies verified)

▼ **Review** ?

Review as: Individual Reviewer

Clarification required on how the sporting goods organization plan to integrate this change with their Web site. Is this going to be a staged roll-out or an over-night switch over?

Attachment:  [Browse...](#)

[Save as Draft](#) [Accept](#) [Reject](#)

▼ **Reviewers** ?

**Individual Reviewers**

Figure 6-4 Sporting Goods Line of Business manager rejects the proposed business solution

If the LoB manager determines that all aspects of the business have been captured in the requirements and charter definitions, the LoB manager can approve the proposal. Figure 6-5 shows the LoB manager providing the confirmation of approval of the business solution charter definition.

The screenshot displays the 'Asset Manager' interface for 'JKHLE'. The navigation bar includes 'Home', 'My Dashboard', 'Communities', 'Assets' (highlighted), and 'Administration'. The main content area is titled 'Asset: Ordering Application [2]' with a help icon. Below this are tabs for 'General Details', 'Content', 'Ratings', 'Forums', 'Review' (selected), and 'Statistics', along with a 'Download' button. The 'Review' section shows 'Solution review 8/3/09' and a 'Review history' link. A message states 'Complete the review and click Accept or Reject.' with a 'Review' button and an 'Actions' dropdown. A 'Policy Compliance Results' section indicates '(1 policies verified)'. The 'Review' section is expanded, showing 'Review as: Individual Reviewer'. A rich text editor contains the text: 'Ordering application business solution definition and charter provide valid definition of the business requirements and case.' Below the editor is an 'Attachment:' field with a 'Browse...' button. At the bottom are three buttons: 'Save as Draft', 'Accept' (highlighted with a red box), and 'Reject'.

Figure 6-5 Sporting Goods LoB manager approves the proposed business solution

## 6.2.2 Application governance review



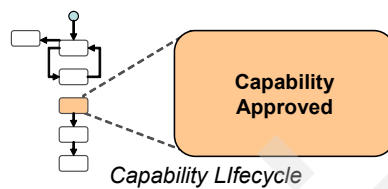
The JKHLE SOA Governance Center of Excellence board receives a notification indicating that the new Ordering application is available for the board's review.

The CoE board meets to discuss whether other business units can benefit from the Ordering application and whether the Ordering application needs to be



enhanced to meet these additional stakeholder needs. In undertaking this review, the board considers both the potential functional and nonfunctional impact of these new stakeholder requirements.

The CoE board decides that the Ordering application is not a duplicate of an existing application. The CoE board also determines that, as an LoB-specific application, the JKHLE Sporting Goods organization must be responsible for the application development and delivery. However, the CoE board decides that in order for the application to be ready for JKHLE's envisaged future expansion into new markets, the application must be National Language-enabled to ensure its reusability. The CoE board representative adds this requirement into the requirements tracking tool so that it is recorded in the definition of the application.



After completing the review of the Ordering application capability, the CoE board approves the creation of the new Business Solution. Figure 6-6 on page 118 shows the CoE board representative's view of the Ordering application. Finally, the CoE board representative confirms that the owning

organization is correctly set by using the asset link to WSRR. This step confirms the CoE board's acceptance of compliance with the governance processes and the CoE board's definition of the application.

The screenshot displays the 'Asset Manager' interface for 'JKHLE'. The top navigation bar includes 'Home', 'My Dashboard', 'Communities', and 'Assets'. The main content area is titled 'Asset: Ordering Application [2]' and features tabs for 'General Details', 'Content', 'Ratings', 'Forums', 'Review' (selected), and 'Statistics'. A 'Download' button is visible in the top right.

Under the 'Review' tab, the section 'Solution review 8/3/09' is active. It includes a 'Review history' link and a welcome message: 'Welcome review board member, evaluate reviewers' reviews and accept or reject asset.' Below this is an 'Evaluate review' button and an 'Actions' dropdown menu.

The 'Policy Compliance Results' section shows '(1 policies verified)'. The 'Evaluate Review' section contains a text area with the comment: 'JKHLE service governance processes followed correctly - Owning organization correctly set etc'. Below the text area are 'Accept' and 'Reject' buttons, with the 'Accept' button highlighted by a red rectangle.

The 'Reviewers' section shows a table of individual reviewers:

Individual Reviewers						
User count	1	Finished reviews	1	Required reviews	1	larry

Below the reviewers table, the 'Reviews' section shows a list of reviews. The first review is by 'Individual Reviewer larry' and is marked as 'Accepted' on 'Aug 3, 2009 5:04 PM'. The review text is: 'Ordering application business solution definition and charter provide valid definition of the business requirements and case.'

Figure 6-6 SOA Governance CoE representative approves the creation of the Ordering application

This approval transitions the business application into the Approved state of the Business Capability lifecycle. The transition to this state provides the final approval that the development of the application can progress.

If the CoE board had identified any issues that required the attention of the business analyst, as with the LoB approval, the CoE board can list their objections as comments and use the Reject option to return the asset to the business analyst for further clarification.

The business analyst has the capability to review the feedback that has been provided during the process by the LoB and CoE representatives, as shown in Figure 6-7.

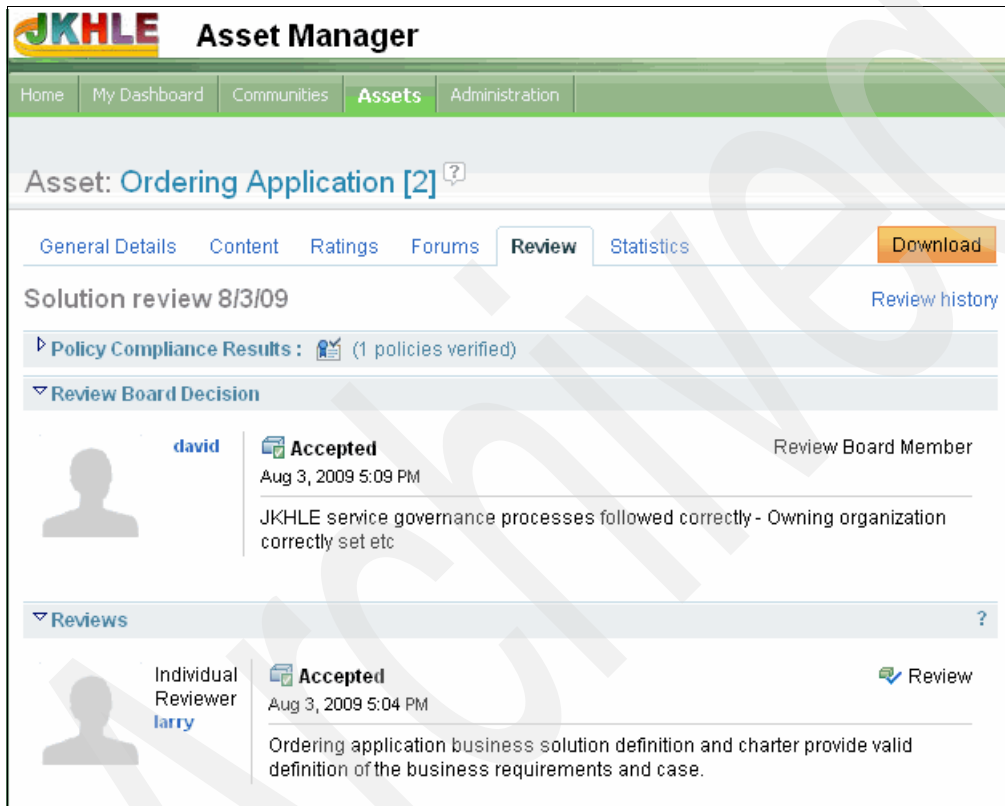


Figure 6-7 Business analyst reviews the feedback

## 6.3 Progress summary

In the first two phases of the process, we have demonstrated how a business analyst can use WSRR Advanced Lifecycle Edition to instigate a process to develop business capabilities when the business analyst identifies a gap in the provided capabilities within the organization.

You have also seen how the SOA Governance Center of Excellence can use the solution as a tool to enforce JKHLE's governance processes.

As can be seen in Figure 6-8, so far we have defined and approved the business case, the business requirements, and the charter. We have also created and completed the definition of the Business Application.

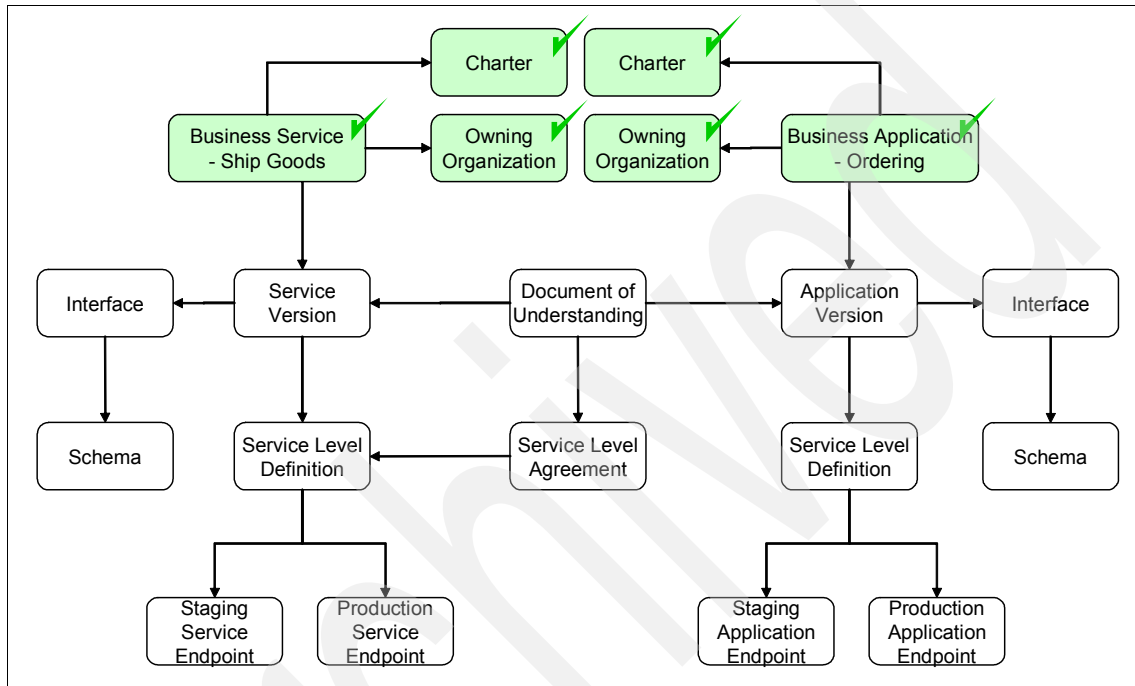


Figure 6-8 Business application and charter as approved entities within our business model

## 6.4 Application scoping and investigation



The Line of Business and Center of Excellence approvals provide the go-ahead for the owning organization to provide funding to proceed with the next steps of the development.

The Sporting Goods organization assigns funding for, and gives the go-ahead to, the business analyst to undertake additional scoping and investigation of the detailed implementation of the proposed Ordering application. During the impact analysis phase of the requirements definition in 5.2, "Business value definition" on page 100, the business analyst identified the business capabilities that exist within the JKHLE organization and that can be reused by the Ordering application. The business analyst now undertakes additional investigation to

determine the specific release of these services, which will be consumed by the new release of the Ordering application.

The business analyst reviews the details of the business capabilities and the details of each release. For each release, the business analyst examines the requirements, as shown in Figure 6-9, and specification details to identify the release that is appropriate to be consumed by the Ordering application.

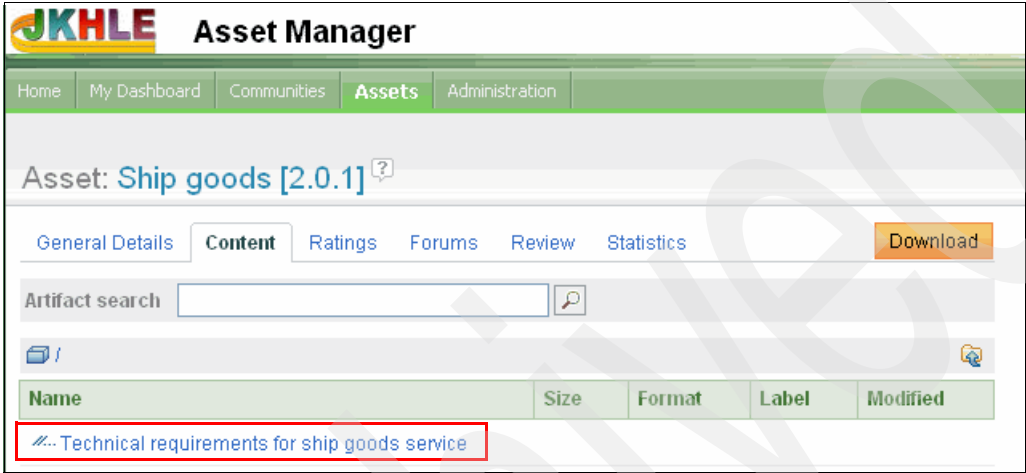
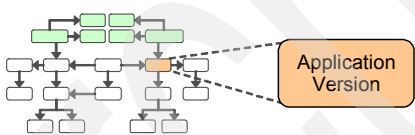


Figure 6-9 Requirements link for the Ship Goods service



During this investigation, the business analyst discovers that the Ship Goods service currently only supports shipment within the U.S. Therefore, the Ship Goods service requires an enhancement in order to support the business requirement to deliver to Canada as well. The business analyst documents the findings by creating a new requirement in JKHLE's requirements tool. Then, the business analyst links this new requirement to a new release of the Ordering application, which the business analyst creates in Rational Asset Manager, as shown in Figure 6-10 on page 122.

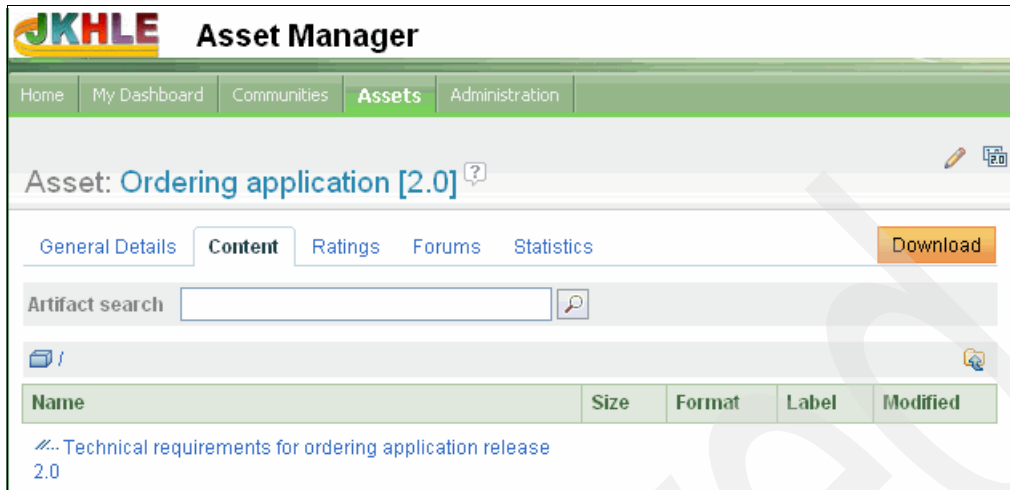


Figure 6-10 New release of the Ordering application in Rational Asset Manager

As part of creating this new release, the business user must classify the asset as being a Web Application (as they did with the business solution) so that it is correctly classified within WSRR. Optionally, they can also associate the release with the business solution that it implements.

**Note:** Currently in WSRR 6.3.0.1, there is a limitation when creating a new release, requiring that the Availability Date be set before saving the asset as a draft. If this date is not set, synchronization of the asset will fail.

When the new release is created in Rational Asset Manager, it is automatically synchronized with WSRR where it is represented as an application version. The business analyst builds a link between the Ordering application Business Capability and this new application version by updating the relationship on the Ordering application business capability, as shown in Figure 6-11 on page 123.

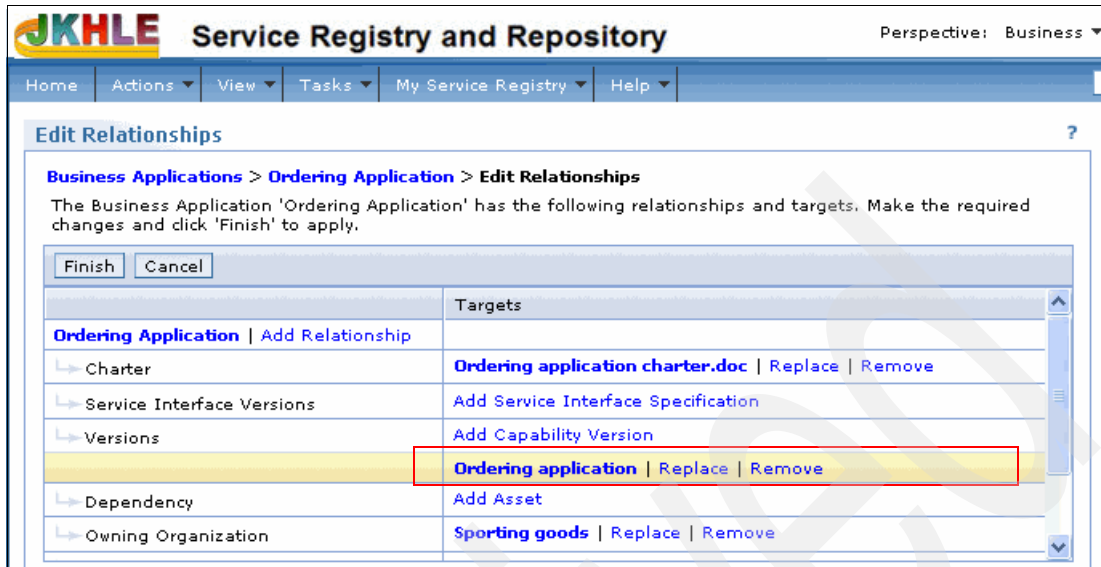
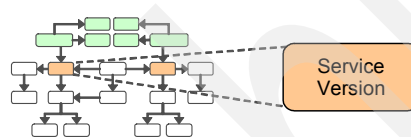


Figure 6-11 Business analyst relates the Ordering application to the Ordering application version



The business analyst documents the required enhancements to the existing Ship Goods service in the JKHLE requirement tool. In order to instigate the process of making these enhancements, the business analyst creates a new release of the Ship Goods service and links this new release to the requirements, as shown in Figure 6-12 on page 124.

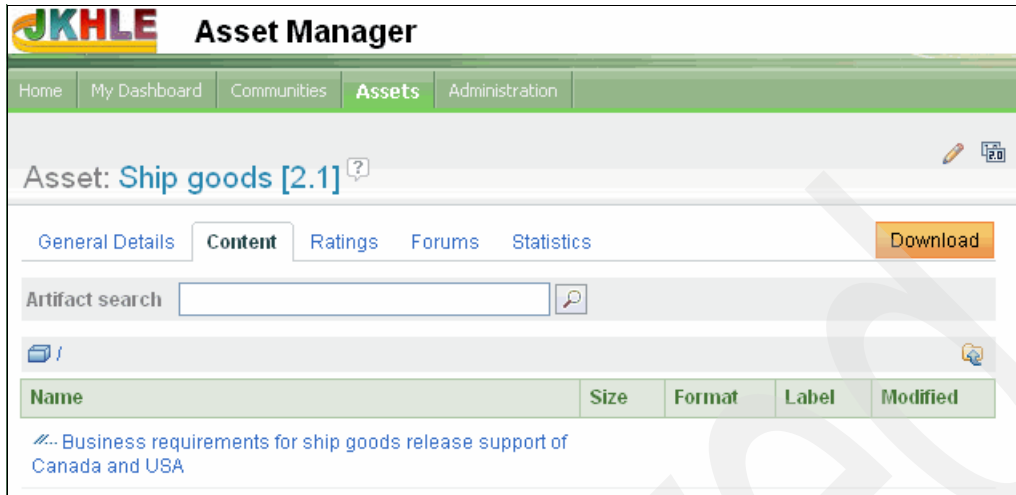


Figure 6-12 New Ship Goods service release

When the business analyst creates the entity, the business analyst ensures that it is correctly classified as an SOA/Service.

As for the Ordering application, the release that is created in Rational Asset Manager is automatically synchronized to WSRR. In the case of the Ship Goods service, this entity is represented as a service version in WSRR. The business analyst builds a link between the Ship Goods Business Capability and this new service version. In WSRR, the business analyst creates the relationship between the new version and the Business Capability that it delivers, as shown in Figure 6-13 on page 125.



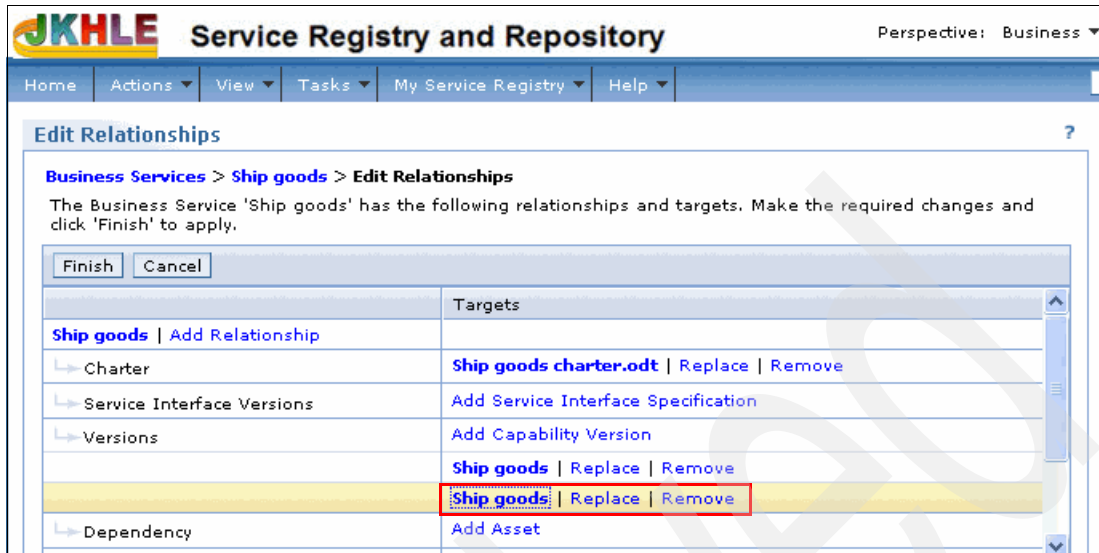
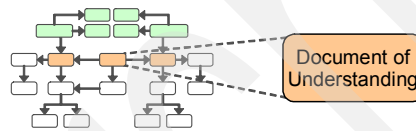


Figure 6-13 Business analyst relates the Ship Goods service to the new Ship Goods service version

## 6.5 Creating Documents of Understanding



In order to instigate the process of defining the business relationship between the Sporting Goods organization and the providers of the three services, the business analyst creates three new

documents of understanding (DOUs) in Rational Asset Manager: one DOU for each of the three provider services that the Ordering application will consume. Each DOU defines the following major features of the relationship:

- ▶ The service version that the application intends to consume
- ▶ The calculated average and peak invocation load (as driven by the business requirements)
- ▶ The business owner of the relationship by specifying the community as being the organization which provides the service

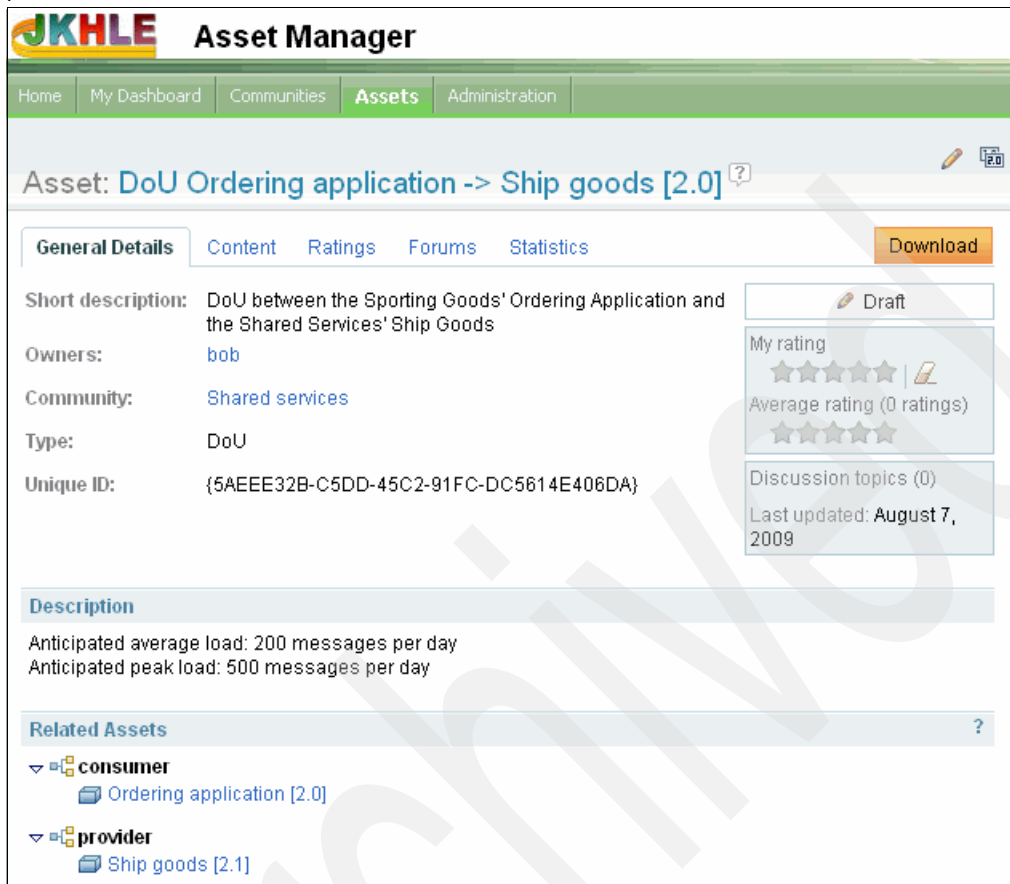


Figure 6-14 New DOU between the Ordering application and the Ship Goods service

**Note:** The business user must now modify the DOU in WSRR in order to specify the owning organization.

## 6.6 Progress summary

At this stage in the process, the business analyst has finalized the business definitions of the new Ordering application. The business analyst has also instigated the development process of the required enhancements to the Ship Goods service by creating the new release of the Ship Goods service, as shown in Figure 6-15 on page 127.

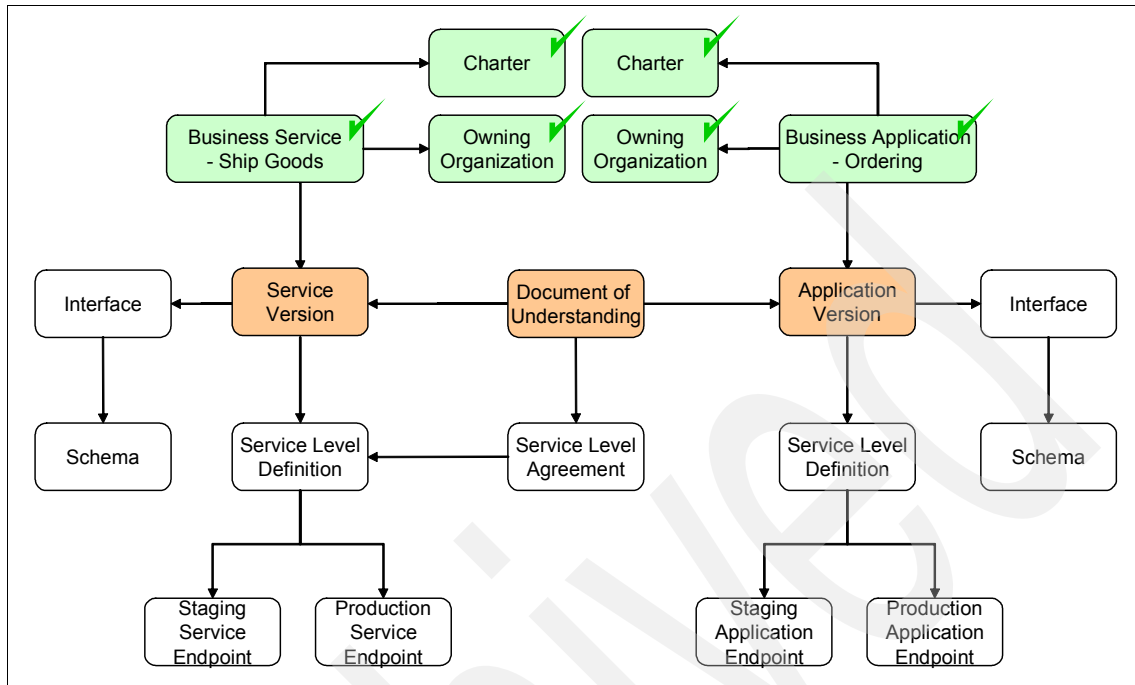


Figure 6-15 New DOUs that have been created in this step of the process



# Release specification

In this chapter, the development organizations define and review the detailed technical specifications of the application and service implementations.

We address these topics in this chapter:

- ▶ 7.1, “Introduction” on page 130
- ▶ 7.2, “Ordering application scope and sizing” on page 130
- ▶ 7.3, “Shipment service scope and sizing” on page 140
- ▶ 7.4, “Development plan business review” on page 151
- ▶ 7.5, “Ship Goods service interface and schema design” on page 152
- ▶ 7.6, “Ordering application interface and schema design” on page 154
- ▶ 7.7, “Document of Understanding proposal” on page 172
- ▶ 7.8, “Service Version specification” on page 179

## 7.1 Introduction

The business analyst has now completed the definition of the application and service requirements and passes responsibility for the continued design elaboration to the development teams. The Sporting Goods development team assumes responsibility for the Ordering application, and the Shared Services development team assumes responsibility for the new release of the Ship Goods service.

## 7.2 Ordering application scope and sizing



The Sporting Goods developer now undertakes the task of elaborating the technical details of the Ordering application version requirements, using the details that are contained in the business application charter and the application version requirements link as the starting point. The development team extends the requirement that was created by the business analyst to detail a number of additional aspects of the implementation:

- ▶ An elaboration of functional capabilities that the implementation will provide
- ▶ A description of the type of business objects that the development team anticipates will be the input and output objects of the application
- ▶ Information about how the application will implement the National Language feature
- ▶ Sizing estimates for each component of the development and test effort

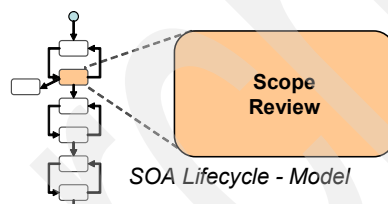
In addition to extending the requirements documentation, the developer also needs to ensure that the application version metadata in WebSphere Service Registry and Repository (WSRR) is complete.

The Ordering application version that has been created in WSRR must have complete and valid metadata in order for it to conform to JKHLE's governance processes. The SOA Governance CoE determines what metadata is required at the various stages of the development process. This metadata is then defined and enforced within the WSRR and Rational Asset Manager configuration.

The developer must specify the following attributes during the design phase:

<b>Name</b>	The name of the application version needs to be modified to include the version number to assist with identification. For the Ordering application, the developer sets this name to Ordering application version 2.0. <sup>5</sup>
-------------	--

<b>Description</b>	The description of the application version needs to be descriptive of the basic functionality that it will deliver. For the Ordering application the developer sets this field to Process a sporting goods order by checking the customer status, take payment from customer and send the order to the warehouse for processing and shipment to the customer.
<b>Owner Email</b>	The Owner Email address must be set to the e-mail address of the individual who will lead the development process. In the case of the Ordering application, the developer sets this address to <code>debra@jkhle.com</code> , the development lead's e-mail address.
<b>Consumer Identifier</b>	The Consumer Identifier is an identifier that the service will pass in the header of all service invocations that it attempts. By passing this identifier, the invoked services will be able to determine whether the consumer has the appropriate Service Level Agreements in place to make the invocation. In the case of the Ordering application, the developer sets this field to <code>SGOA020</code> .
<b>Owning Organization</b>	The Owning Organization relationship must be populated with a link to the Sporting Goods organization.



Scope Review, and triggers a notification to the SOA Governance CoE indicating the need for their input.

After completing the elaboration of the scope and providing an estimated sizing, the developer proposes the application version for scope review, as shown in Figure 7-1 on page 132. This step places the Ordering application version into the next stage of the service-oriented architecture (SOA) lifecycle,

<sup>5</sup> The name is changed in WSRP so that when the entity is a target of a relationship, it is clear to users what version it is. Rational Asset Manager automatically displays the version of the entity in brackets after the name.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Application Version

Application Versions > Ordering application version 2.0

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application version 2.0

Description

Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

Version

2.0

Consumer Identifier

SGOA020

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0

Remote State

Draft

Owner Email

debra@jkhle.com

Additional Properties

Back

New Version

Propose Scope

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

None

Provides

None

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ordering Application

Consumer DOU(s)

DoU Ordering Application -> Customer Status Check

Figure 7-1 Sporting Goods developer proposes the scope of the Ordering application version

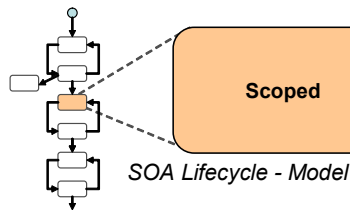
132 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition



## 7.2.1 Ordering application scope review



The SOA Governance CoE representative, having been notified of the application version's transition to the Scope Review state, reviews the scope and sizing that have been provided. The CoE representative verifies that the scope that has been specified by the developer delivers the complete solution as defined by the business charter and requirements. The CoE representative ensures that the provided sizings are appropriate for the amount of effort that is required to deliver each component of the solution.



To approve the scope and sizing, the CoE representative transitions the Ordering application version to the Scoped state in the SOA lifecycle, as shown in Figure 7-2 on page 134.

JKHLE

Service Registry and Repository

Perspective: SOA Governance ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Application Version

Application Versions > Ordering application version 2.0

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application version 2.0

Description

Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

Version

2.0

Consumer Identifier

SGOA020

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&w=2.0

Remote State

Draft

Owner Email

debra@jkhle.com

Additional Properties

Back

New Version

Revise Scope

Approve Scope

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

None

Provides

None

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ordering Application

Consumer DOU(s)

DoU Ordering Application -> Customer Status Check

DoU Ordering Application -> Execute Payment

Figure 7-2 SOA Governance CoE representative approves the scope of the Ordering application version

134 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition

## 7.2.2 Ordering application development plan



The final stage of this part of the process is for the development team to create a development plan. The Sporting Goods development lead creates this document through consultation with the relevant developers and testers, and ensures that it contains the following key components:

- ▶ Functionality to be delivered
- ▶ User stories that will be realized
- ▶ Design overview
- ▶ Integration points
- ▶ Work item breakdown with development and test sizings
- ▶ Limitations with the final deliverable
- ▶ Stretch goals for implementation if there is time

The development team associates the development plan with the Ordering application version by attaching it as an artifact using WSRR, as shown in Figure 7-3 on page 136.

**Service Registry and Repository**
Perspective: Development ▾

[Home](#)
[Actions ▾](#)
[View ▾](#)
[Tasks ▾](#)
[My Service Registry ▾](#)
[Help ▾](#)

### Application Version

**Application Versions > Ordering application version 2.0**

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

[Details](#)
[Impact Analysis](#)
[Governance](#)
[Policy](#)
[Activity](#)

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

**Properties**

- \*Name  
Ordering application version 2.0
- Description  
Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for
- Version  
2.0
- Consumer Identifier  
SGOA020
- Version Availability Date  
Tuesday, 1 September 2009
- Version Termination Date  
Tuesday, 4 August 2009
- Version Requirements Link  
[urn:serviceregistry](#)
- Asset Web Link  
[https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0](#)
- Remote State  
Draft
- Owner Email  
[debra@jkhle.com](#)

**Links**

- [Graphical View](#)
- [Applied Policies](#)
- [Applied Policy Attachments](#)

**Relationships**

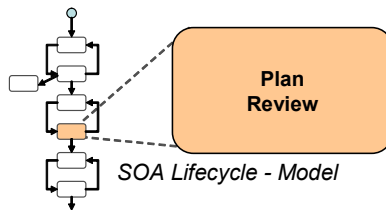
- Interface Specifications  
None
- Provided Web Services  
None
- Provided SCA Modules  
None
- Owning Organization  
[Sporting goods](#)
- Dependency  
None
- Aggregation  
None
- Artifacts  
[Ordering application development plan.doc](#)
- Provides  
None
- Consumes  
None

**Dependent Entities**

- Chartered Business Capability(s)  
[Ordering Application](#)
- Consumer DOU(s)  
[DoU Ordering Application -> Customer Status Check](#)

[Back](#)
[New Version](#)
[Propose Plan](#)
[New SLD](#)

Figure 7-3 Sporting Goods development lead attaches the development plan to the Ordering application version



The development lead then proposes this plan to the release manager and the business representatives by transitioning the application version to the Plan Review state, as shown in Figure 7-4 on page 137.

**Service Registry and Repository**

Perspective: Development | don

Home | Actions | View | Tasks | My Service Registry | Help

### Application Version

**Application Versions > Ordering application version 2.0**

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details | Impact Analysis | Governance | Policy | Activity

Edit Properties | Edit Relationships | Edit Classifications

**Properties**

- \*Name: Ordering application version 2.0
- Description: Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for
- Version: 2.0
- Consumer Identifier: SGOA020
- Version Availability Date: Tuesday, 1 September 2009
- Version Termination Date: Tuesday, 4 August 2009
- Version Requirements Link: urn:service:registry
- Asset Web Link: https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0
- Remote State: Draft
- Owner Email: debra@jkhle.com

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

- Interface Specifications: None
- Provided Web Services: None
- Provided SCA Modules: None
- Owning Organization: Sporting goods
- Dependency: None
- Aggregation: None
- Artifacts: Ordering application development plan.doc
- Provides: None
- Consumes: None

**Dependent Entities**

- Chartered Business Capability(s): Ordering Application
- Consumer DOU(s): DoU Ordering Application -> Customer Status Check

**Additional Properties**

Back | New Version | **Propose Plan** | New SLD

Figure 7-4 Sporting Goods development lead proposes the plan for the Ordering application version

### 7.2.3 Ordering application development plan review



With the Ordering application scope and sizing defined and the development plan complete, the LoB manager must now review the work of the development team to ensure that the LoB manager agrees with their specification, assumptions, and sizings. The the LoB manager can add additional reviewers to the process to get input on areas that are out of the LoB manager's sphere of expertise. In JKHLE's Rational Asset Manager configuration, the release manager is automatically set as an additional reviewer. The release manager must provide input on the development process aspects of the specification, as well as consider the resourcing aspects of the project delivery.

Using Rational Asset Manager, the release manager can review the complete specification and, upon completion of that review, provide approval of the development plan, as shown in Figure 7-5 on page 139. This approval is an indication that the development team has sufficient resources available to deliver the application solution. Additionally, the release manager is also confirming agreement with the development team that the scope, sizing, and development plan are accurate and complete.

**JKHLE Asset Manager**

Home My Dashboard Communities **Assets**

Search Results >

Asset: **Ordering application [2.0]**

General Details Content Ratings Forums **Review** Statistics [Download](#)

**Release review 8/4/09** [Review history](#)

Complete the review and click Accept or Reject. [Review](#) [Actions](#)

**Policy Compliance Results :** (1 policies verified)

**Review**

Review as: Individual Reviewer

The development plan looks accurate and complete. I am happy that there is sufficient development resource for this to proceed.

Attachment:  [Browse...](#)

[Save as Draft](#) [Accept](#) [Reject](#)

Figure 7-5 Sporting Goods release manager approves the development plan of the Ordering application version

Because of the Ordering application's dependence on the Ship Goods service, the LoB representative withholds approval of the development plan while the Ship Goods service specification is defined. The Ordering application review remains in the In Review state in Rational Asset Manager. The business review steps will occur after the detailed development process steps have been created for the Ship Goods service.

## 7.3 Shipment service scope and sizing



The Shared Services developer now undertakes the task of elaborating the technical details of the Ship Goods service version requirements, taking the details contained in the service version requirements link as their starting point. The development team extends the requirements that were created by the business analyst to detail a number of additional aspects of the implementation:

- ▶ An elaboration of functional capabilities that the implementation will provide
- ▶ A description of the type of business objects that the development team anticipates will be the input and output objects of the application
- ▶ Sizing estimates for each component of the development and test effort

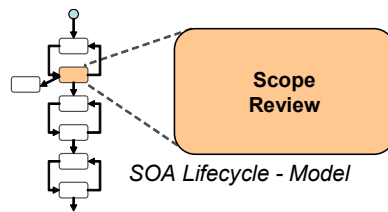
In addition to extending the requirements documentation, the developer also needs to ensure that the service version metadata in WSRR is complete.

The Ship Goods service version that has been created in WSRR must have complete and valid metadata in order for it to conform to the JKHLE's governance processes. The developer must specify the following attributes during the design phase:

<b>Name</b>	The name of the service version must be modified to include the version number to assist with identification. For the Ship Goods service, the developer enters Ship goods version 2.1.
<b>Description</b>	The description of the service version needs to depict the basic functionality that it will deliver. For the Ship Goods service, the developer types Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada.
<b>Owner Email</b>	The Owner Email address must be set to the e-mail of the person who is leading the development process. In the case of the Ship Goods service, the developer enters connie@jkhle.com, the development lead's e-mail address.
<b>Consumer Identifier</b>	The Consumer Identifier is an identifier that the service will pass in the header of all service invocations that it attempts. By passing this identifier, the invoked services will be able to determine whether the consumer has the appropriate Service Level Agreements in place in order to make the invocation. In the case of the Ship Goods service, it is not anticipated that the service will be a consumer; however, JKHLE's governance processes



require that this field is set. The developer types SSSG021 for the Consumer Identifier.



After completing the elaboration of the scope, the developer proposes the service version for scope review, as shown in Figure 7-6 on page 142. This step places the Ship Goods service version into the next stage of the SOA lifecycle, Scope Review, and triggers a notification to the development lead indicating the need for the development lead's input.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Draft

Owner Email

connie@jkhle.com

Additional Properties

Back

New Version

Propose Scope

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

None

Provides

None

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Provider DOU(s)

None

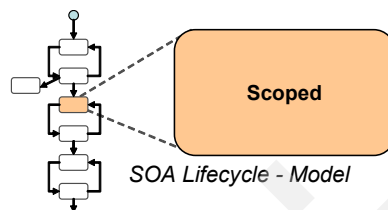
Figure 7-6 Shared Services developer proposes the scope of the Ship Goods service version

### 7.3.1 Shipment service scope review



The SOA Governance Center of Excellence (CoE), having been notified of the service version's transition to the Scope Review state, reviews the scope that has been provided. The CoE verifies that the scope that has been specified by the developer delivers the complete solution as defined by the requirements.

The SOA Governance CoE also reviews the development plan to ensure that it follows the internal development processes, verifying compliance with code review and conformance to any legal requirements, such as the use of any external code, for example, open source or third-party libraries.



To approve the scope, the CoE representative transitions the Ship Goods service to the Scoped state in the lifecycle, as shown in Figure 7-7 on page 144.



### 7.3.2 Ship Goods service development plan



The final stage of this part of the process is for the development team to create a development plan. The Shared Services development lead creates this document and ensures that it contains the following key components:

- ▶ Functionality to be delivered
- ▶ User stories that will be realized
- ▶ Design overview
- ▶ Integration points
- ▶ Breakdown of the work items with development and test sizings
- ▶ Limitations with the final deliverable
- ▶ Stretch goals for implementation if there is time

The Shared Services development lead then associates the development plan with the Ship Goods service version by attaching it as an artifact using WSRR, as shown in Figure 7-8 on page 146.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Draft

Owner Email

connie@jkhle.com

Additional Properties

Back

New Version

Propose Plan

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

None

Consumes

None

Dependent Entities

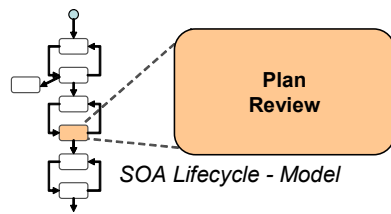
Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Figure 7-8 Shared Services developer attaches the development plan to the Ship Goods service version



The developer then proposes this plan to the release manager and the business representatives by transitioning the service version to the Plan Review state, as shown in Figure 7-9 on page 148.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Draft

Owner Email

connie@jkhle.com

Additional Properties

Back

New Version

Propose Plan

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

None

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Provider DOU(s)

Figure 7-9 Shared Services developer proposes the plan for the Ship Goods service version



### 7.3.3 Ship Goods service development plan review



With the Ship Goods service scope and sizing defined and the development plan complete, the line of business (LoB) must now review the work of the development team to ensure that the LoB agrees with the development team's specification, assumptions, and sizings. The line of business can add additional reviewers to the process to get input on areas that are out of the LoB's sphere of expertise. In JKHLE's Rational Asset Manager configuration, the release manager is automatically designated as an additional reviewer. The release manager must provide input on the development process aspects of the specification, as well as consider the resourcing aspects of the project delivery.

Using Rational Asset Manager, the Shared Services release manager can review the complete specification, and upon completion of that review, provide approval of the development plan, as shown in Figure 7-10 on page 150. This approval is an indication that the development team has sufficient available resources to deliver the service solution. Additionally, the Shared Services release manager is also confirming agreement with the development team that the scope, sizing, and development plan are accurate and complete.

**JKHLE Asset Manager**

Home My Dashboard Communities **Assets**

Search Results >

Asset: **Ship goods [2.1]** ?

General Details Content Ratings Forums **Review** Statistics **Download**

**Release review 8/4/09** [Review history](#)

Complete the review and click Accept or Reject. [Review](#) **Actions** v

▸ **Policy Compliance Results :** (1 policies verified)

▾ **Review** ?

Review as: Individual Reviewer

The development plan has been reviewed and appears to be complete and accurate. We have checked the resources available for development and are happy that there is sufficient bandwidth.

Attachment:  **Browse...**

**Save as Draft** **Accept** **Reject**

Figure 7-10 Shared Services release manager approves the plan for the Ship Goods service version

JKHLE's governance process review step requires approval from both the release manager and the LoB representative, as described in 4.2.7, "Governance review process" on page 90. The business review step can now occur.

## 7.4 Development plan business review



After completing the scope, sizing, and development plan specification for both the Ordering application and the Ship Goods service, the business can now review the proposals. The business analyst reviews both of the development plans to ensure that they fulfill the business case and requirements that were originally specified. If the business analyst determines that the requirements have all been satisfied, the business analyst can approve both of the development plans in Rational Asset Manager. Figure 7-11 shows the business analyst approving the development plan for the Ordering application.

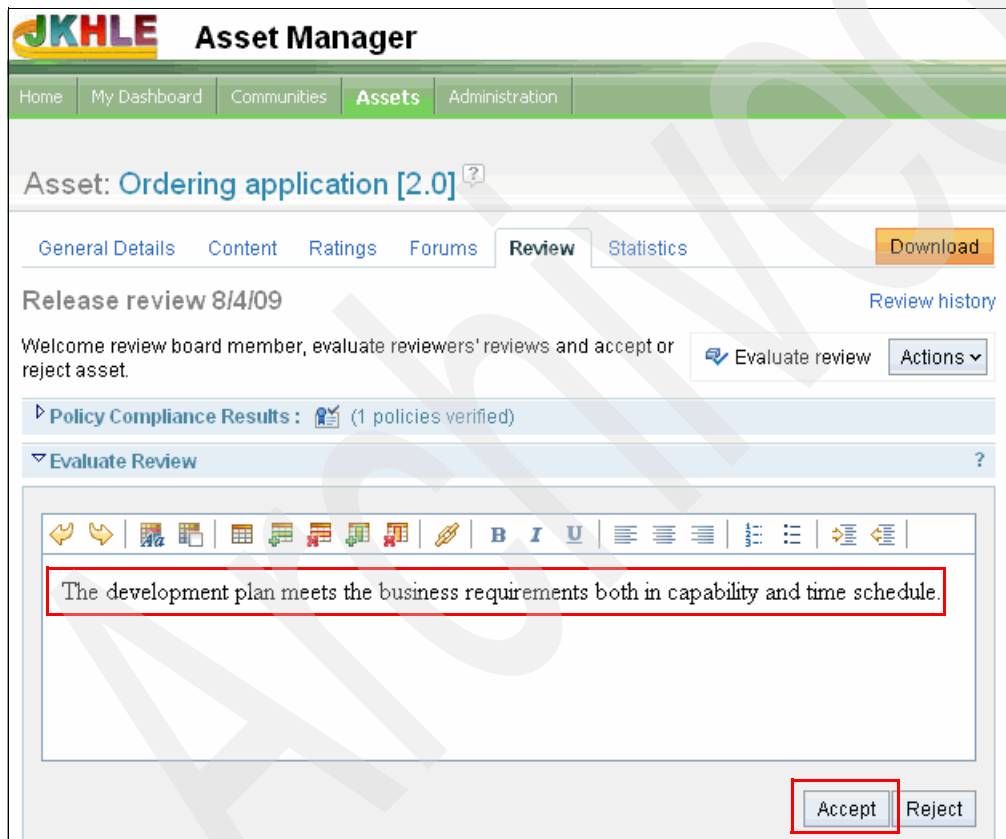


Figure 7-11 Business analyst accepts the plan for the Ordering application version

If the business analyst has any concerns or requires any modifications to the plan, the business analyst can reject the plan, as shown in Figure 7-12 on page 152, with comments that indicate which areas need modification. The development team can then work to incorporate any changes or enhancements

as required and then return the plan to the business analyst and the release manager for another plan review cycle.

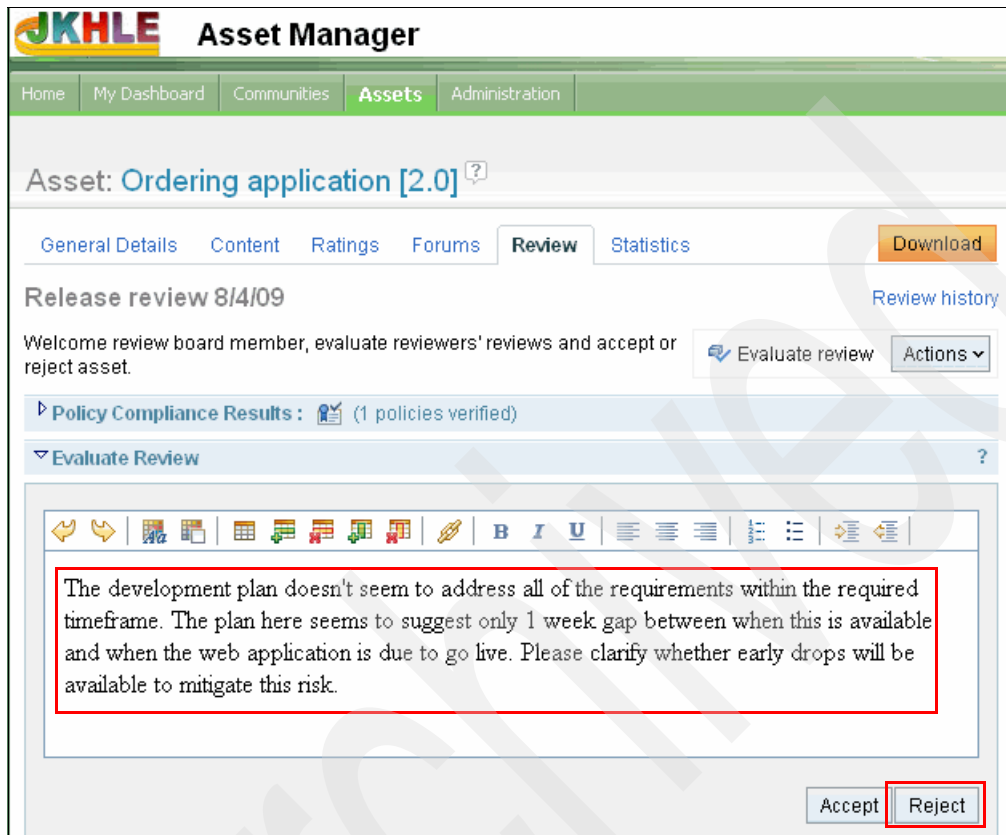


Figure 7-12 Business analyst rejects the plan for the Ordering application version

## 7.5 Ship Goods service interface and schema design



The Shared Services development lead is notified that the service version plan has been approved, and the team can begin the actual development process. The first stage of this development process is to design the interfaces, schemas, and business objects that will be used when interacting with this new version of the Ship Goods service. The developer searches the registry for existing interfaces, schemas, and business object definitions that are suitable for reuse within the Ship Goods service scenario. The developer determines that the existing Ship Goods service interface and schema definitions will fulfill the requirements for the new service version, which eliminates the need for undertaking any design review cycle for the interface and schema assets

themselves. The developer creates the relationship to the existing service interface and schema specifications, as shown in Figure 7-13 on page 153.



Figure 7-13 Shared Services developer relates the existing interface with the new Ship Goods service version

Figure 7-14 on page 154 shows the interface and schema specifications that the developer has linked to the new service version. Note that these interface and schema specifications are already approved, and therefore, they are shown as complete in this view of the modeled entities.

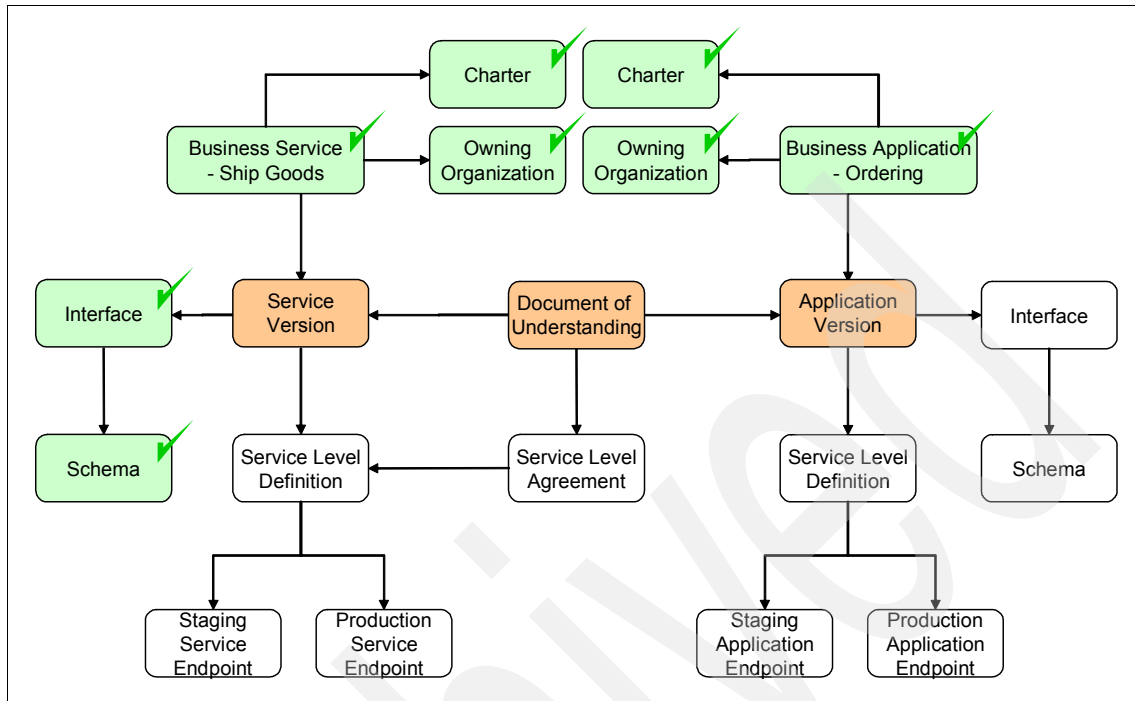
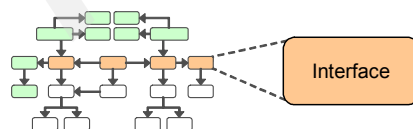


Figure 7-14 Current state of the business modeled entities for the Ship Goods service

## 7.6 Ordering application interface and schema design



The Sporting Goods development lead is notified that the application version plan has been approved, and the team can begin the actual development process. The first stage of this development process is to design the interfaces, schemas, and business objects that will be used when interacting with this new version of the Ordering application. The developer searches the registry for existing interfaces, schemas, and business object definitions that are suitable for reuse within the Ordering application scenario. The developer determines that the existing business object definitions, which are used for the three services which the application is to consume, can form the basis of the new schema specification.



The developer creates a new interface specification under which all of the definition details will be collated; it is this high-level entity that will be governed. The service interface specification defines a

particular interaction pattern and message structure that is supported across the realized versions of the business application. As the development process progresses, this structure will be elaborated to reference the Web Services Description Language (WSDL) port type that forms the basic definition of the interaction mechanism for the application. Figure 7-15 shows the developer creating the Service Interface Specification in Rational Asset Manager.

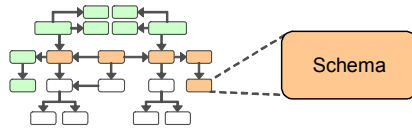
The screenshot shows the Rational Asset Manager web application. The top navigation bar includes links for Home, My Dashboard, Communities, Assets, and Administration. The main heading is 'Submit "Ordering application interface specification"'. Below this is a progress bar with four steps: Describe, Attach, Categorize, and Associate. The 'Describe' step is currently active. A message states 'Verify the asset information is correct.' The 'Asset Information' section contains the following details:

Name:	Ordering application interface specification
Version:	2.0
Community:	Sports department
Type:	Specification
Short description:	The interface specification that will denote the messages that can be sent to the ordering application

Below the asset information is the 'Categories' section, which shows a single category: 'Software Development' with the description 'SOA: Specification / Service / Interface'. This category is highlighted with a red box. The 'Related Assets' section is empty. At the bottom right, there are four buttons: 'Back', 'Save as Draft' (highlighted with a red box), 'Submit As Is', and 'Submit'.

Figure 7-15 Sporting Goods developer creates a Service Interface Specification and saves it as a draft


**Note:** The categorization of the service interface and the schema specifications is critical to the synchronization between WSRP and Rational Asset Manager. Ensure that assets are categorized correctly, as detailed in 2.8.3, “Service Interface Specification, Specification” on page 58 and 2.8.4, “Schema Specification, Specification” on page 59.



The developer then creates a new schema specification that will collate the definition details of the interface message definitions. This high-level definition is used to collate the various artifacts that define the

schema. Figure 7-16 on page 157 shows the developer creating the new schema specification in Rational Asset Manager. Note the detailed information that the developer provides to describe how to design the schema.




**Asset Manager**

[Home](#)
[My Dashboard](#)
[Communities](#)
[Assets](#)
[Administration](#)

### Submit "Ordering application schema specification" ?

Describe
Attach
Categorize
Associate
Complete

Verify the asset information is correct.

#### Asset Information Modify asset

Name:	Ordering application schema specification
Version:	2.0
Community:	Sports department
Type:	Specification
Short description:	The schema specification that will denote the structure of the messages that can be sent to the ordering application
Namespace:	http://www.jkhle.com/governance/schema/oa
Target Namespace:	No Artifact met the criteria of the constraint
Description:	<p>This will be composed of three business objects; customer account, payment and shipment. Each of these business objects will be a complex type. Customer account will contain just name and account number. Payment will contain card details (number, expiry and card security code) and billing address. Shipment will contain recipients name and address. The same address complex type will be reused across the payment and shipment objects.</p>

#### Categories Modify asset

Software Development	SOA: Specification / Schema
----------------------	-----------------------------

#### Related Assets Modify asset

[Back](#)
[Save as Draft](#)
[Submit As Is](#)
[Submit for Review](#)

Figure 7-16 Sporting Goods developer creates a Schema Specification and saves it as a draft

**Important:** It is critical that the Namespace is set correctly on a schema specification before it is put into the Review state in Rational Asset Manager. Failure to do so will cause a break in the synchronization. It is possible to configure Rational Asset Manager to verify this setting, and we recommend that this verification is configured by the systems administrator.

After creating the specification assets in Rational Asset Manager, the developer then needs to create the relationships between the application version and these two new assets. In WSRR, the developer first creates a relationship between the Ordering application version and the interface specification, as shown in Figure 7-17.

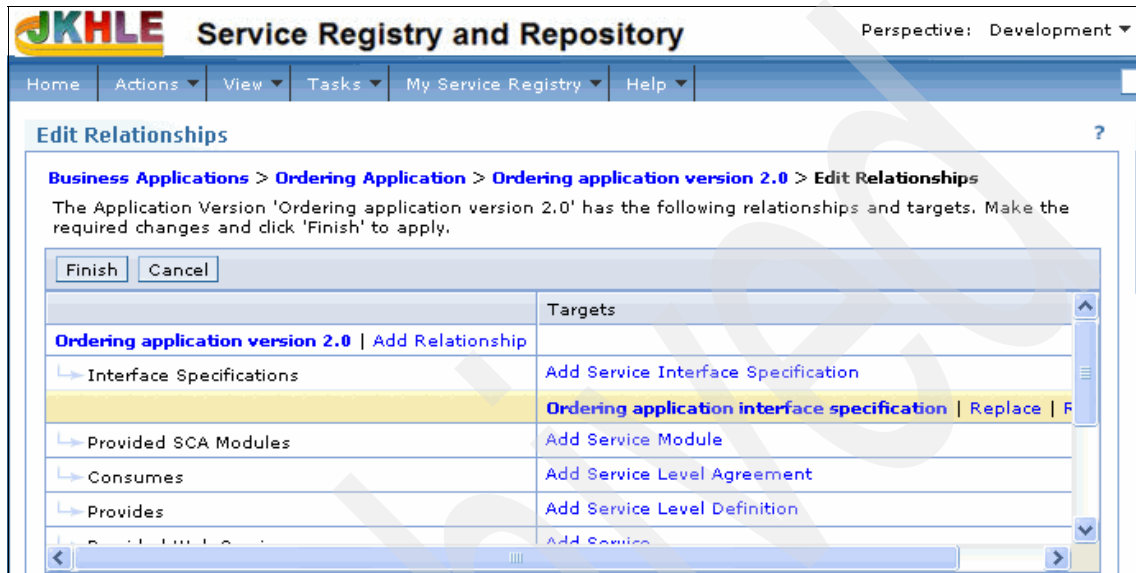


Figure 7-17 Sporting Goods developer creates the relationship between the application version and the interface specification

The Sporting Goods developer then creates the relationship from the interface specification to the schema specification, as shown in Figure 7-18 on page 159.

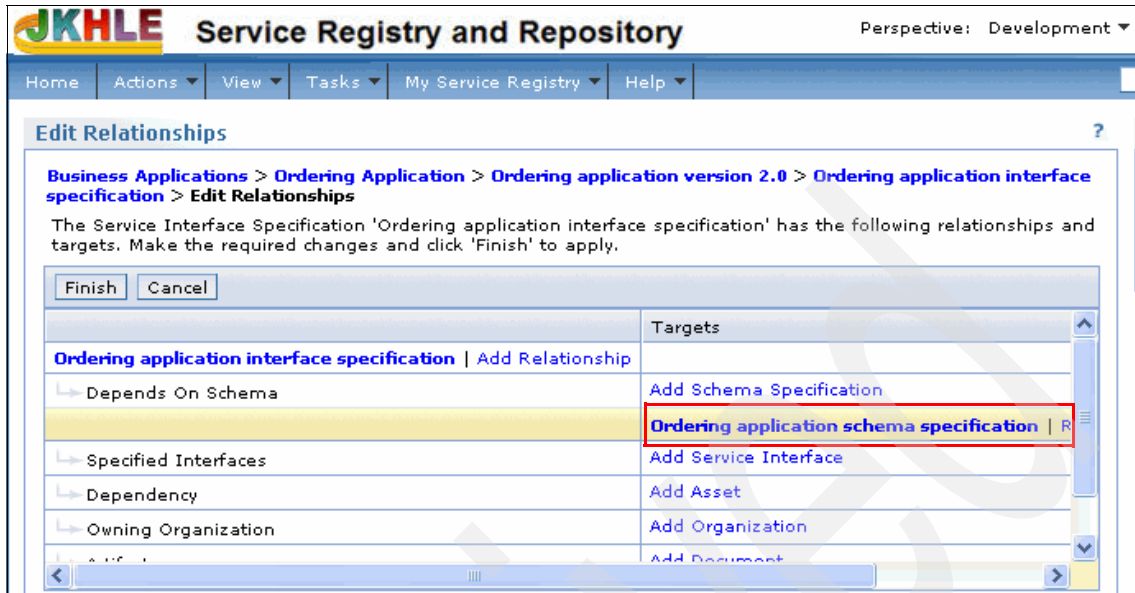
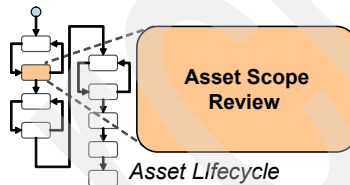


Figure 7-18 Sporting Goods developer creates the relationship between the interface and the schema specifications

We also recommend that the developer assigns the owning organization to both the interface and schema specifications at this time.



After defining the scope of the two specifications, the developer now needs to propose both of these specifications for Scope Review. Figure 7-19 on page 160 shows the developer proposing the scope of the interface specification.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Service Interface Specification

Business Applications > Ordering Application > Ordering application version 2.0 > Ordering application interface specification

Detail view for Service Interface Specification. The Service Interface Specification defines a particular interaction pattern and message structure that is supported across the realized versions of business capabilities. This is independent of any transport or quality of service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application interface specification

Description

The interface specification that will denote the messages that can be sent to the ordering application

Version

2.0

Version Namespace

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B7846D40B-84FC-4E3F-2235-3C10A49A3C25%7D&v=2.0

Remote State

Draft

Owner Email

Additional Properties

Back

Propose

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Depends On Schema

Ordering application schema specification

Specified Interfaces

None

Owning Organization

None

Dependency

None

Aggregation

None

Artifacts

None

Dependent Entities

Dependent Business Capability(s)

None

Dependent Capability Version(s)

Ordering application version 2.0

Figure 7-19 Sporting Goods developer proposes the interface specification

## 7.6.1 Ordering application interface and schema review

The SOA Governance CoE representative reviews the details of the schema and interface specifications to ensure that the developer has correctly defined their scope. The primary purpose of this review is to ensure that the design maximizes the reuse of existing assets within the organization. Therefore, the review in

160 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition

this case focuses on the proposed business object and interface reuse. The CoE representative also ensures that the namespace is defined correctly, because this namespace must be the same as the namespace of the document that implements this schema specification.

To confirm the acceptance of the schema and interface scopes that have been specified, the CoE representative approves both of the proposals. Figure 7-20 shows the CoE representative approving the Ordering application schema specification. This approval step must be repeated for the service interface specification, as well.

**JKHLE Service Registry and Repository** Perspective: SOA Governance ▾

Home ▾ Actions ▾ View ▾ Tasks ▾ My Service Registry ▾ Help ▾

### Schema Specification

[Service Interface Specifications](#) > [Ordering application interface specification](#) > [Ordering application schema specification](#)

Detail view for Schema Specification. A Schema Specification Asset defines shared schema documents (XSD) allowing them to be governed independently.

Details | Impact Analysis | Governance | Policy | Activity

[Edit Properties](#) | [Edit Relationships](#) | [Edit Classifications](#)

**Properties**

**\*Name**  
Ordering application schema specification

**Description**  
The schema specification that will denote the structure of the messages that can be sent to the ordering application

**Namespace**  
<http://www.jkhle.com/governance/schema/oa>

**Version**  
2.0

**Asset Web Link**  
<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BCF0448AE-EDB8-162C-B001-B74C4F0FE623%7D&w=2.0>

**Remote State**  
Draft

**Owner Email**

**Links**

- [Graphical View](#)
- [Applied Policies](#)
- [Applied Policy Attachments](#)

**Relationships**

Specified Schemas  
None

Depends on Schema  
None

Owning Organization  
None

Dependency  
None

Aggregation  
None

Artifacts  
None

**Governance State**  
Asset Scope Review

**Classifications**  
Schema Specification  
Asset Scope Review

**Additional Properties**

[Back](#) [Reject Proposal](#) [Approve Proposal](#)

Figure 7-20 SOA Governance CoE representative approves the scope of the schema specification

## 7.6.2 Ordering application interface and schema artifact development



After the scope of the interface and schema specifications has been approved, the development team can begin creating the artifacts that realize the high-level specifications. The developer for the Ordering application creates a new XML Schema Definition (XSD) document in which the developer defines the schema specification within Rational Software Architect. This new schema specification defines the message objects that will be used by the Ordering application. Simultaneously, the developer creates a WSDL document to define the interface that uses the schema document as the basis for the interface's message definition.

As the developer iterates through versions of the schema and interface designs, the developer loads them into Rational Asset Manager using the Eclipse plug-in within the Rational Software Architect tool, as shown for the schema document in Figure 7-21.

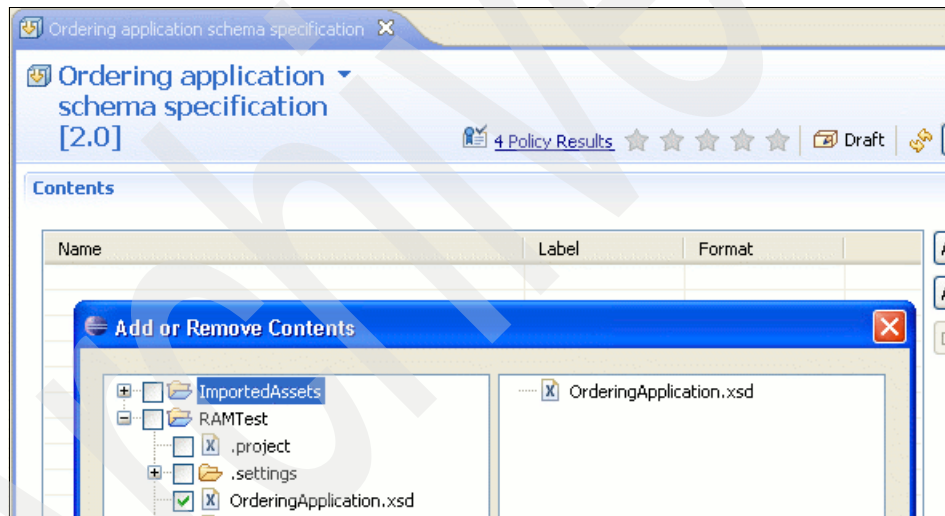
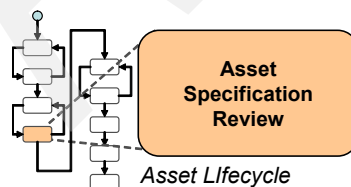


Figure 7-21 Sporting Goods developer loads schema specification from an Eclipse-based tool



When the developer has completed the design of the interface and schema realizations, the developer proposes both the interface and schema specification for review by the SOA Governance CoE and the Sporting Goods release manager, as shown (for the interface specification) in Figure 7-22 on page 163.

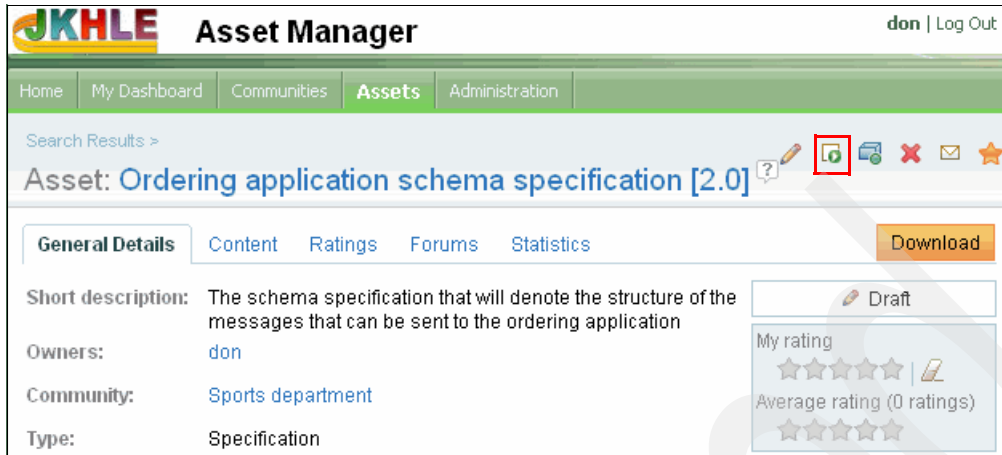


Figure 7-22 Sporting Goods developer proposes the service interface specification in Rational Asset Manager

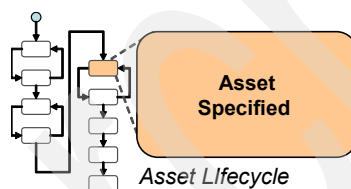
### 7.6.3 Ordering application interface and schema review



The SOA Governance CoE adds the Sporting Goods release manager to the review process for both the schema and interface specifications, as shown in Figure 7-23 on page 164.



Figure 7-23 SOA Governance CoE representative adds the required reviewers for the service interface proposal of the Ordering application version



The Sporting Goods release manager reviews the interface and schema specification realizations that have been defined by the Sporting Goods development team. The Sporting Goods release manager verifies that the definitions follow the JKHLE's development guidelines and ensures that they fulfill the

high-level definition of the specifications. To confirm the acceptance of these definitions, the release manager approves both of the specifications by transitioning them to the Approved state in Rational Asset Manager, as shown in the example in Figure 7-24 on page 165.





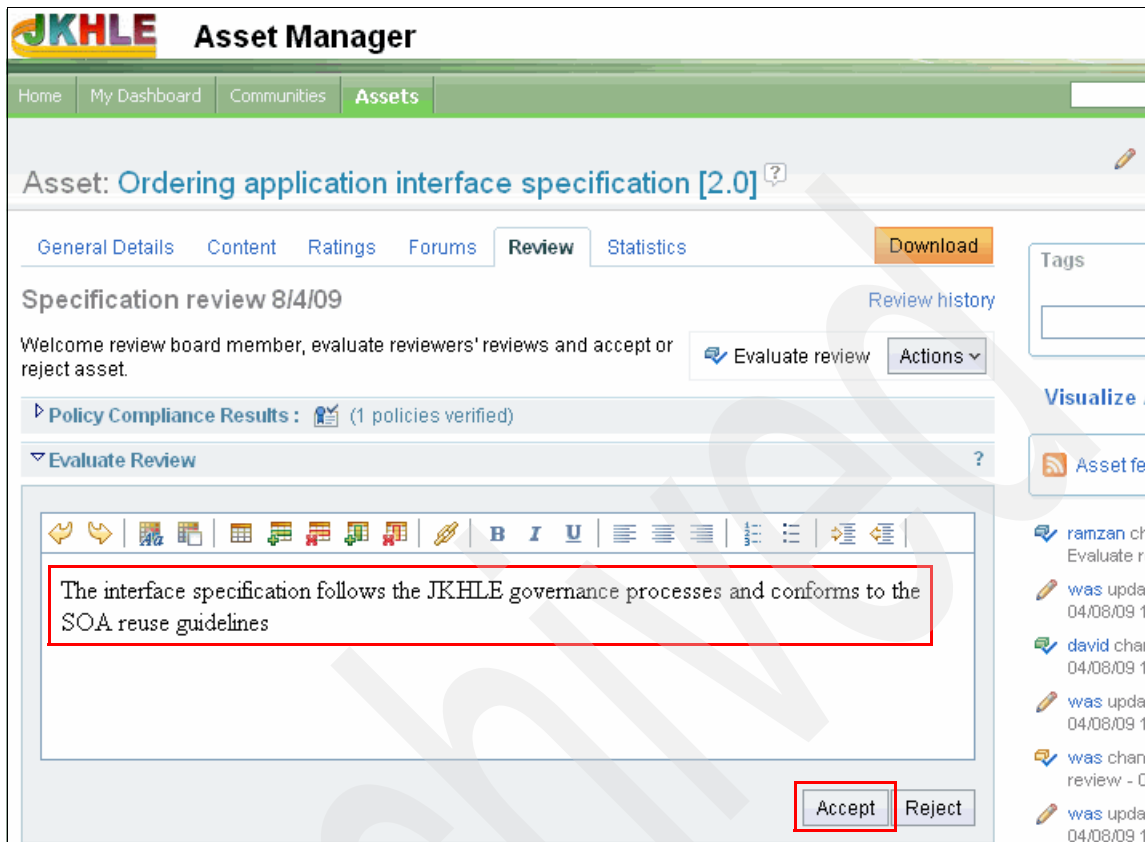


Figure 7-25 SOA Governance CoE representative accepts the proposed interface specification for the Ordering application version

## 7.6.4 Specification relationships



When the schema specification is approved in Rational Asset Manager, the Sporting Goods developer can publish the appropriate specification documents to WSR. The developer opens the specification definition in Rational Asset Manager and clicks the Publish icon, which is highlighted in Figure 7-26 on page 167, to start the process.

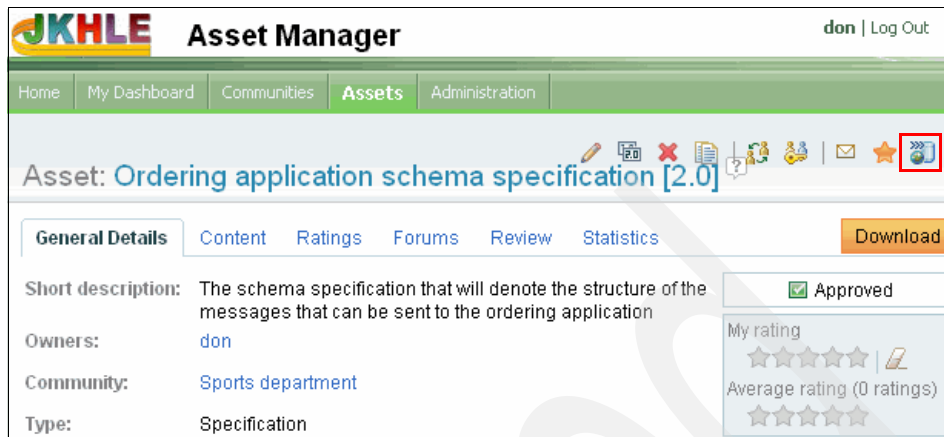


Figure 7-26 Sporting Goods developer publishes artifacts to WSRR

The developer selects the version of the document to publish to WSRR, as shown in Figure 7-27.

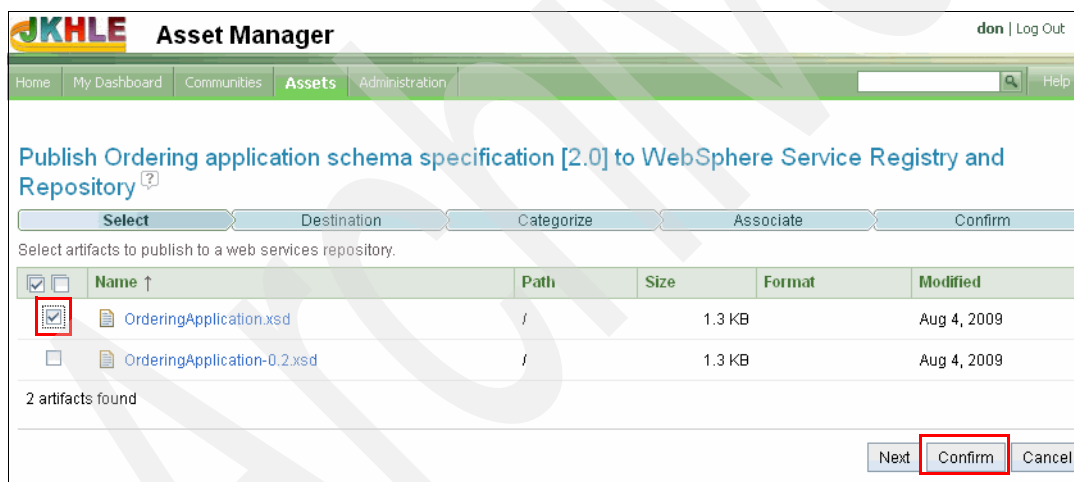
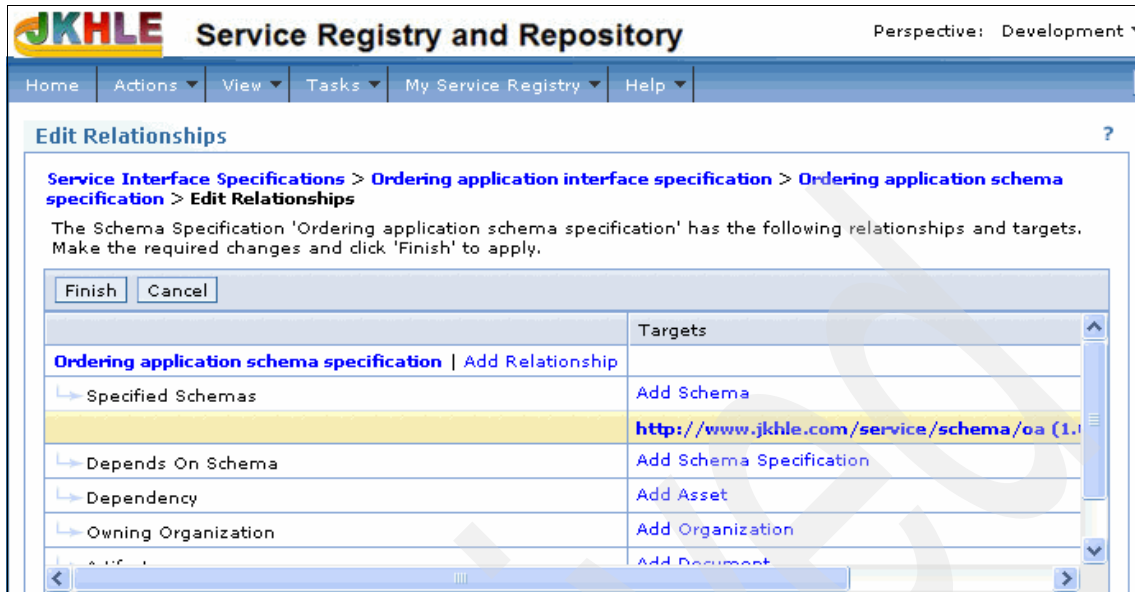


Figure 7-27 Sporting Goods developer selects the document to publish to WSRR

The developer executes this step for both the schema and interface specification documents. These published documents will automatically be associated as an artifact of the specification entities in WSRR, which are now in the Specified state of the Asset lifecycle.

The developer relates the specification entities to the schema that is defined within the document artifacts, as shown in Figure 7-28 on page 168.



*Figure 7-28 Sporting Goods developer creates the relationship from the schema specification to the specification artifact that is defined in the schema document loaded in the previous step*

The developer repeats this step for the interface specification, relating the interface specification to the interfaces that are specified in the interface specification artifact, as shown in Figure 7-29 on page 169.

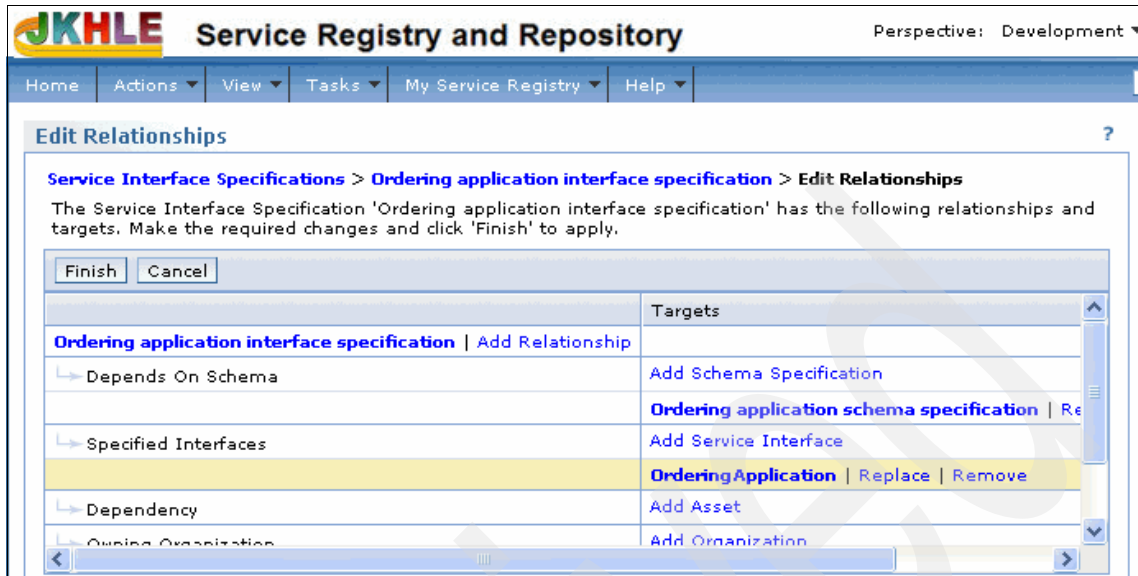
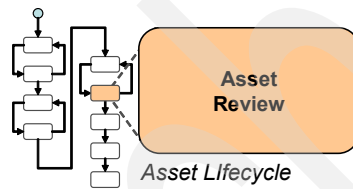


Figure 7-29 Sporting Goods developer relates the WSDL interface document to the interface specification



The developer proposes these relationships for review by the Sporting Goods release manager, as shown in Figure 7-30 on page 170.

JKHLE

Service Registry and Repository

Perspective: Development ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Schema Specification

Schema Specifications > Ordering application schema specification

Detail view for Schema Specification. A Schema Specification Asset defines shared schema documents (XSD) allowing them to be governed independently.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application schema specification

Description

The schema specification that will denote the structure of the messages that can be sent to the ordering application

Namespace

<http://www.jkhle.com/service/schema/oa>

Version

2.0

Asset Web Link

<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BB4AF8D95-7:BC3D-4162-4F3B3556E856%7D&v=2.0>

Remote State

Approved

Owner Email

debra@jkhle.com

Additional Properties

Back

Submit Asset For Approval

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Specified Schemas

<http://www.jkhle.com/service/schema/oa> (1.0)

Depends on Schema

None

Owning Organization

None

Dependency

None

Aggregation

None

Artifacts

[OrderingApplication.xsd](#)

Governance State

Asset Specified

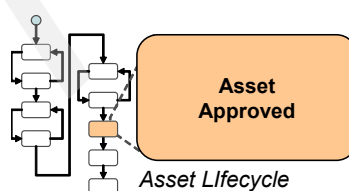
Classifications

Asset Specified

Schema Specification

Figure 7-30 Sporting Goods developer submits the schema for approval

## 7.6.5 Specification asset review



The final step in the process is for the Sporting Goods release manager to review the specified relationships that the developer previously specified for the interface and schema specifications. After verifying that the relationships have been built correctly, the

release manager approves the specifications, as shown for the schema specification in Figure 7-31.

JKHLE

Service Registry and Repository

Perspective: Development ▾ |

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Schema Specification

Schema Specifications > Ordering application schema specification

Detail view for Schema Specification. A Schema Specification Asset defines shared schema documents (XSD) allowing them to be governed independently.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application schema specification

Description

The schema specification that will denote the structure of the messages that can be sent to the ordering application

Namespace

<http://www.jkhle.com/service/schema/oa>

Version

2.0

Asset Web Link

<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BB4AF8D95-7BC3D-4162-4F3B3556E856%7D&v=2.0>

Remote State

Approved

Owner Email

debra@jkhle.com

Additional Properties

Back

Revise Asset

Approve

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Specified Schemas

<http://www.jkhle.com/service/schema/oa> (1.0)

Depends on Schema

None

Owning Organization

None

Dependency

None

Aggregation

None

Artifacts

[OrderingApplication.xsd](#)

Governance State

Asset Review

Classifications

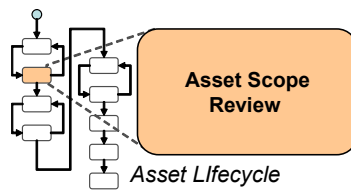
Schema Specification

Asset Review

Figure 7-31 Sporting Goods release manager approves the schema specification

Chapter 7. Release specification 171

## 7.7 Document of Understanding proposal



With all of the interfaces and the schemas now approved, the developer proposes the three documents of understanding to the provider organizations. This proposal triggers a business-level negotiation between the consumer and provider organizations to create an agreement to provide the requested service.

Figure 7-32 on page 173 shows the developer proposing the DOU between the Ordering application and the Ship Goods service. The developer repeats this step for the Customer Status Check and Execute Payment services.



JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

DOU

DOUs > DoU Ordering application -> Ship goods

Detail view for DOU. A DOU or Document of Understanding represents an agreement between two Organizations whereby they agree how the consuming Organizations realization of their capability (Capability Version) will use the providing organizations realized capability.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

Name

DoU Ordering application -> Ship goods

Description

DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B31C44DE8-2-BDDAA3924632%7D&w=2.0

Remote State

Draft

Owner Email

Requirements Link

urn:serviceregistry

Additional Properties

Back

Propose

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Consumer

Ordering application version 2.0

Provider

Ship goods

Subscribed Service Level Agreements

None

Owning Organization

None

Aggregation

None

Dependency

None

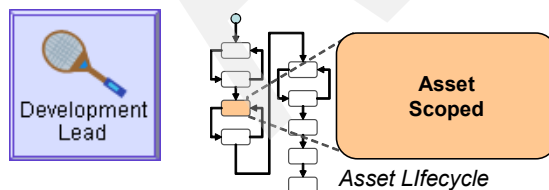
Artifacts

None

Governance State

Figure 7-32 Sporting Goods developer proposes a DOU between the Ordering application version and the Ship Goods service version

## 7.7.1 Document of Understanding scope review



owning organization.

The Sporting Goods development lead is notified of the proposed DOUs and reviews the technical details of the required relationship. The Sporting Goods development lead ensures that the developer has correctly specified the intended providers, as well as followed the JKHLE governance guidelines in specifying the

After reviewing the technical details of the DOUs, the development lead approves their scope, as shown in Figure 7-33.

**JKHLE Service Registry and Repository** Perspective: Development ▾

Home ▾ Actions ▾ View ▾ Tasks ▾ My Service Registry ▾ Help ▾

**DOU**

**DOUs > DoU Ordering application -> Ship goods**

Detail view for DOU. A DOU or Document of Understanding represents an agreement between two Organizations whereby they agree how the consuming Organizations realization of their capability (Capability Version) will use the providing organizations realized capability.

Details | Impact Analysis | Governance | Policy | Activity

Edit Properties | Edit Relationships | Edit Classifications

**Properties**

\*Name  
DoU Ordering application -> Ship goods

Description  
DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

Asset Web Link  
<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B31C44DE8-2BD0AA3924632%7D&v=2.0>

Remote State  
Draft

Owner Email

Requirements Link  
<urn:serviceregistry>

**Additional Properties**

Back | Reject Proposal | **Approve Proposal**

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

Consumer  
Ordering application version 2.0

Provider  
Ship goods

Subscribed Service Level Agreements  
None

Owning Organization  
Shared services

Aggregation  
None

Dependency  
None

Artifacts  
None

**Governance State**

Figure 7-33 Sporting Goods development lead approves the proposal of the Document of Understanding

## 7.7.2 Document of Understanding development review

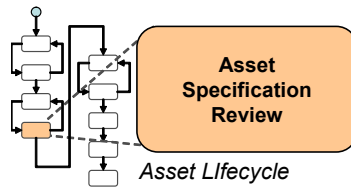


The Sporting Goods release manager reviews the Documents of Understanding that the development lead has proposed and elaborates on these DOUs by creating a requirement in the JKHLE requirement tool to specify the business-level requirements of the relationship, including:

- The proposed details for the development funding, specifically identifying which organization will fund any necessary pieces of development. For

example, specify that the Sporting Goods organization will fund the development of the new Ship Goods service version.

- Specify the proposed charge-back mechanism and unit invocation costs.
- Specify the timetables that are required in order to meet their business commitments.



The release manager adds the link to the requirement in the DOU definition. Then, after reviewing and elaborating the DOUs, the release manager proposes the plan for business review, as shown in Figure 7-34 on page 175.

**Service Registry and Repository**

Perspective: Development | ramzan

[Home](#) | [Actions](#) | [View](#) | [Tasks](#) | [My Service Registry](#) | [Help](#)

## DOU

**DOUs > DoU Ordering application -> Ship goods**

Detail view for DOU. A DOU or Document of Understanding represents an agreement between two Organizations whereby they agree how the consuming Organizations realization of their capability (Capability Version) will use the providing organizations realized capability.

[Details](#) | [Impact Analysis](#) | [Governance](#) | [Policy](#) | [Activity](#)

[Edit Properties](#) | [Edit Relationships](#) | [Edit Classifications](#)

**Properties**

**\*Name**

DoU Ordering application -> Ship goods

**Description**

DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

**Asset Web Link**

<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B31C44DE8-2BD0AA3924632%7D&v=2.0>

**Remote State**

Draft

**Owner Email**

**Requirements Link**

<http://reqs.jkhle.com/req-928>

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

**Consumer**

Ordering application version 2.0

**Provider**

Ship goods version 2.1

**Subscribed Service Level Agreements**

None

**Owning Organization**

Shared services

**Aggregation**

None

**Dependency**

None

**Artifacts**

None

**Additional Properties**

[Back](#) | [Propose Asset Specification](#)

**Governance State**

Figure 7-34 Sporting Goods release manager proposes the DOU plan

### 7.7.3 Document of Understanding business review

After the Documents of Understanding have been approved by the development team, they are made available for review by the business stakeholders. JKHLE has defined the following set of required approvals to approve the business relationship:

<b>Release Manager</b>	The Shared Services release manager must review each DOU that has been created for which the Shared Services development team is the provider.
<b>Operations</b>	The Shared Services operations manager must review the proposed usage levels to ensure that sufficient resources are available to service the anticipated load.
<b>Business Unit Leader</b>	The Shared Services Line of Business manager must approve the funding and charge-back mechanisms and levels that have been proposed in each DOU. Because the Shared Services business unit is the provider organization, this DOU is an agreement to the levels of charge-back that are proposed and an agreement to the level of reimbursement for the Ship Goods service enhancements.
<b>Business Unit Leader</b>	The Sporting Goods Line of Business manager must approve the funding and charge-back mechanisms that have been proposed in each DOU. Because the Sporting Goods business unit is the consumer organization, this DOU is an agreement to fund the development of the enhancements to the Ship Goods service and an agreement to future charges for consuming the three services.
<b>CoE</b>	The CoE must approve the proposed relationship to ensure that it complies with the JKHLE governance processes.



The Shared Services LoB manager defines the individuals, who are required to perform the review, and then starts the review process, as shown in Figure 7-35 on page 177.



**Asset Manager**

[Home](#)
[My Dashboard](#)
[Communities](#)
[Assets](#)
[Administration](#)

**Asset: DoU Ordering application -> Ship goods [2.0]**

[General Details](#)
[Content](#)
[Ratings](#)
[Forums](#)
[Review](#)
[Statistics](#)
[Download](#)

**DoU review 8/4/09**
[Review history](#)

Welcome review board member, evaluate reviewers' reviews and accept or reject asset.
 [Evaluate review](#)
[Actions](#)

**Policy Compliance Results :** (1 policies verified)

**Evaluate Review**

The shared service team accept the funding and charge-back proposition.  
 Having verified the reviewer's feedback we accept this DoU as proposed.

[Accept](#)
[Reject](#)

**Reviewers**

**Individual Reviewers**

	david	Review	Accepted
	jon	Review	Accepted
	larry	Review	Accepted
	simon	Review	Accepted

**Reviews**

Individual Reviewer  
**simon**

**Accepted**  
 Aug 4, 2009 4:03 PM  
 The development team have the available resource to meet the requested schedule

Review

Figure 7-36 Shared Services LoB manager gives final approval

## 7.8 Service Version specification

In order to complete the technical aspects of the specification of the Ordering application and the Ship Goods service, the developer must define the binding protocols and endpoints that will be exposed. Both the Ship Goods service and the Ordering application are to be exposed as Web services. Therefore, in order to provide this final specification, the developer must create a WSDL defining the binding, port, and namespace definition of the endpoints. This specification completes the technical design aspects of the application and services. It also allows the development process of any consuming applications to continue by confirming the detailed implementation definition that is necessary to create any invoking code.

**Leading practice:** Although this step in the process requires the definition of a WSDL that will create an endpoint definition, this definition will be a temporary value that is used only for specification definition. It will never be invoked.

This binding and endpoint definition is a requirement of the JKHLE service governance process. The WSRR Governance Enablement Profile does not enforce that this definition is defined at this stage in the lifecycle.

### 7.8.1 Ship Goods endpoint specification



The Shared Services developer creates the WSDL documents that define the binding and ports that complete the service specification. The Shared Services developer loads these documents into WSRR, as shown in Figure 7-37 on page 180.



Figure 7-37 Shared Services developer loads WSDL documents defining the binding and specification endpoints for the service interface

After loading the documents into WSRR, the developer needs to complete the step by relating the Ship Goods service to the definition that was just loaded. The Shared Services developer edits the relationships of the Ship Goods service version and builds the link to the specification endpoint definition that was loaded in the previous step, as shown in Figure 7-38.

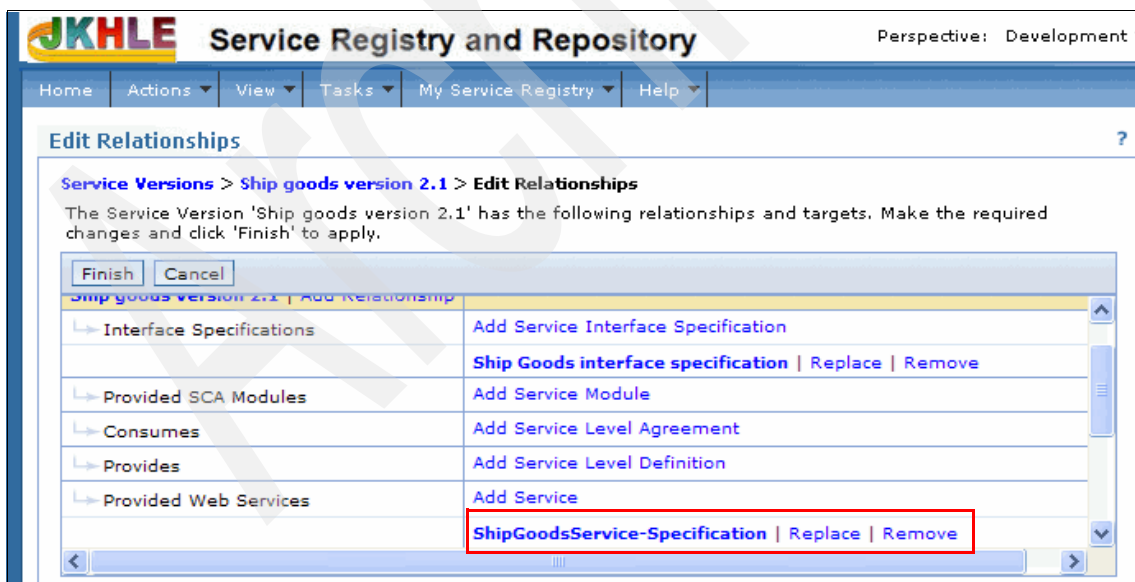


Figure 7-38 Shared Services developer relates the Ship Goods service version to the specification endpoints



## 7.8.2 Ordering application endpoint specification



The Sporting Goods developer needs to complete the same step that the Shared Services developer just completed. The Sporting Goods developer creates the WSDL documents that define the binding and ports that complete the service specification. The Sporting Goods developer loads these documents into WSRR, as shown in Figure 7-39.



Figure 7-39 Sporting Goods developer loads WSDL documents defining the binding and specification endpoints for the service interface

After loading the documents into WSRR, the Sporting Goods developer needs to complete the step by relating the Ordering application to the definition that was just loaded. The Sporting Goods developer edits the relationships of the Ordering application version and builds the link to the specification endpoint definition that was loaded in the previous step, as shown in Figure 7-40 on page 182.

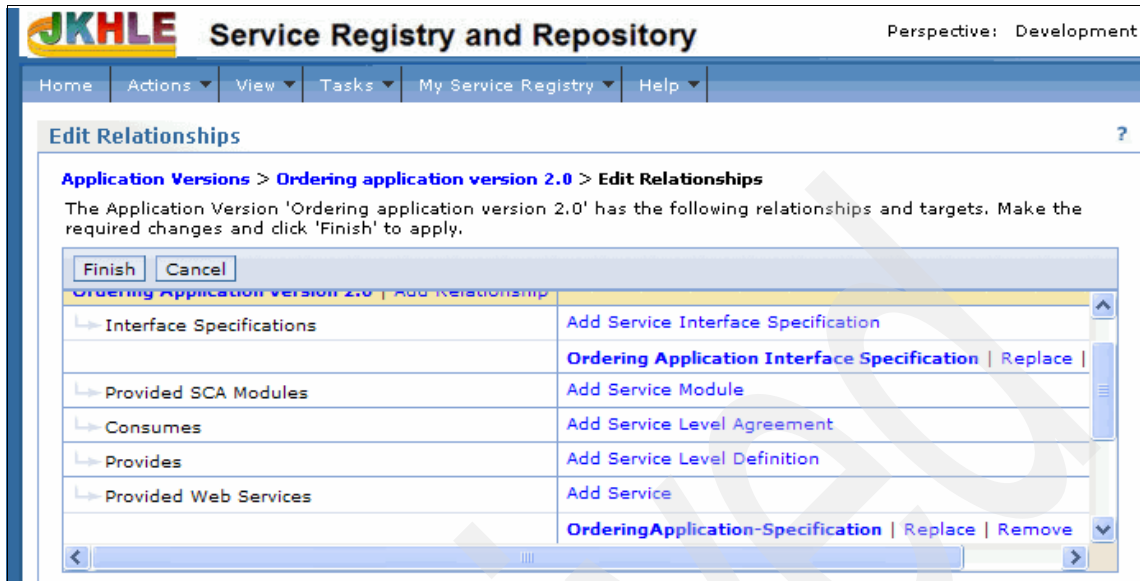


Figure 7-40 Sporting Goods developer relates the Ordering application version to the specification endpoints

## 7.9 Progress summary

Figure 7-41 on page 183 shows that at this stage of the process the development teams have completed the specification of the Ordering application and Ship Goods service interfaces and schemas. Also, the two organizations have agreed to the required consumer/provider relationships.

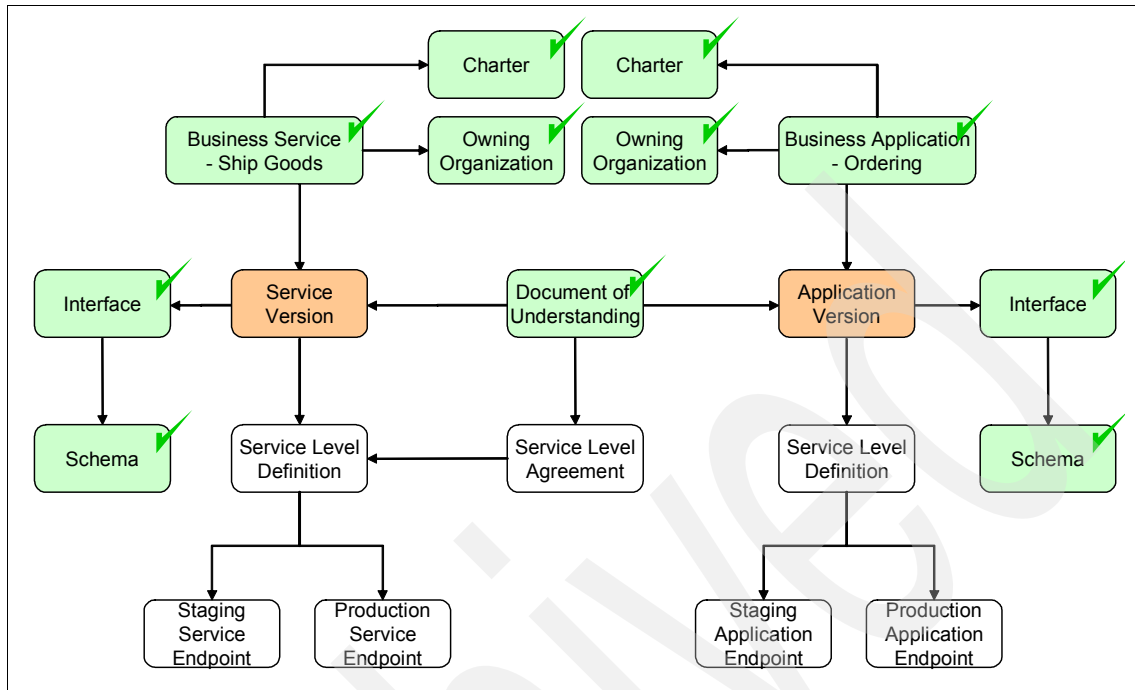


Figure 7-41 Ordering application and Ship Goods service interfaces, schemas, and DOUs have been completed



# Defining Service Level Agreements

In this chapter, the application and service development teams define the service levels that they will provide to their consumers. The Sporting Goods development team then creates the service level agreements with the services that the Ordering application will consume.

This chapter includes these topics:

- ▶ 8.1, “Introduction” on page 186
- ▶ 8.2, “Creating a Service Level Definition” on page 186
- ▶ 8.3, “Creating Service Level Agreements” on page 199
- ▶ 8.4, “Version specification approval” on page 208

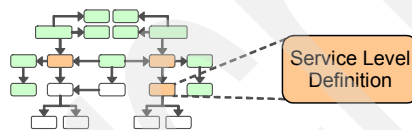
## 8.1 Introduction

The previous chapters focused on defining the business requirements and developing the technical specifications. We then used the capabilities of the WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition to enforce a review and approval process. Now that the development team has completed the technical specifications, the final step in the version specification process is to define the Service Level Definitions that will be provided and consumed.

## 8.2 Creating a Service Level Definition

The Sporting Goods and Shared Services teams need to define the service levels that they will provide to their consumers. In WSRR, a *service level* is defined in a Service Level Definition (SLD) entity. A single application or service version might provide a number of SLDs to allow consumer services to select the quality of service (QoS) that they require. The Ordering application and the Ship Goods service will provide only a single level of QoS.

### 8.2.1 Ordering application Service Level Definition



The Sporting Goods developer reviews the details of the business QoS requirements, as detailed in the business case and requirements documentation on the application version specification. The

Sporting Goods developer then uses this information to create a new SLD to define the level of service that will be made available to the Web site, as well as the interface that must be used to interact with the application, which can be seen in Figure 8-1 on page 187.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Definition

Application Versions > Ordering application version 2.0 > SLD - Ordering application version 2.0

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLD - Ordering application version 2.0

Description

Defines the standard QoS that will be provided by the Ordering application to the Web site

Average Response Time

5 seconds

Availability

24/7 High Availability

Additional Properties

Back

Propose Scope

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

OrderingApplication

Available Endpoints

None

Bound SCA Export

None

Bound Web Service Port

None

Compatible Service Level Definitions

None

Dependent Entities

Consuming Service Level Agreement(s)

None

Providing Capability Version

Ordering application version 2.0

Governance State

SLD Identified

Classifications

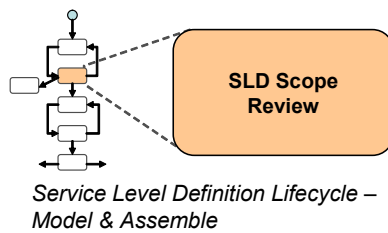
SLD Identified

Extended Service Level Definition

Figure 8-1 Sporting Goods developer creates an SLD for the Ordering application version

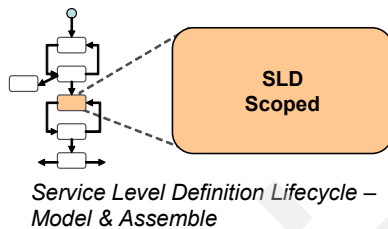
**Note:** In order to ensure that the SLD has all required properties, we recommend that you use the **New SLD** button on the Details view of the Capability Version to create the entity.

Chapter 8. Defining Service Level Agreements 187



The developer then proposes this SLD specification for review by the service-oriented architecture (SOA) Governance Center of Excellence (CoE).

## 8.2.2 Ordering application Service Level Definition scope review



The CoE representative reviews the Service Level Definition that has been specified in order to verify that the proposed qualities of service will meet the business requirements. Additionally, the CoE representative reviews the definition to ensure that it adheres to the JKHLE governance process standards. The CoE representative approves the defined

scope, as shown in Figure 8-2 on page 189.





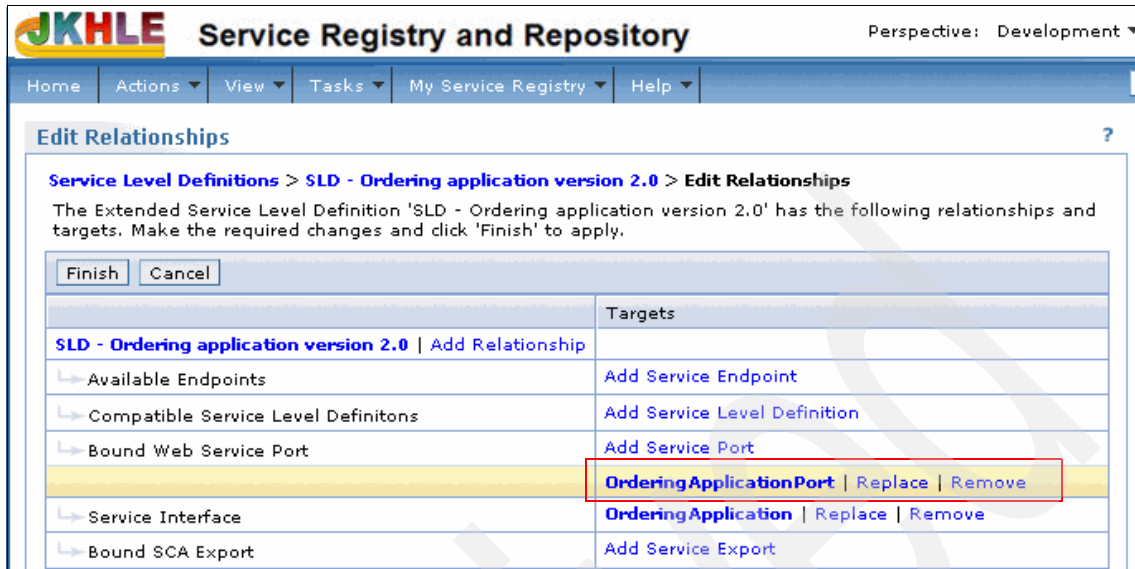
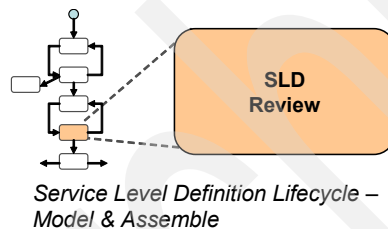


Figure 8-3 Sporting Goods developer relates the SLD to the specified ports



After completing the specification, the developer can propose this specification for review by the CoE, as shown in Figure 8-4 on page 191.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Definition

Service Level Definitions > SLD - Ordering application version 2.0

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLD - Ordering application version 2.0

Description

Defines the standard QoS that will be provided by the Ordering application to the Web site

Average Response Time

5 seconds

Availability

24/7 High Availability

Additional Properties

Back

Propose Specification

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

OrderingApplication

Available Endpoints

None

Bound SCA Export

None

Bound Web Service Port

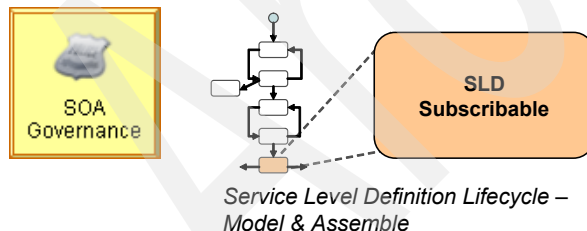
OrderingApplicationPort

Compatible Service Level Definitions

None

Figure 8-4 Sporting Goods developer proposes the specification of the SLD

## 8.2.4 Ordering application Service Level Definition review



The CoE representative reviews the service port specification that has been proposed. They verify that the specified endpoints will provide the bindings suitable for the application consumers and that JKHLE guidelines are being followed, specifically around the security aspects of the specified protocol. The CoE representative approves the specification, as shown in Figure 8-5 on page 192.

JKHLE

Service Registry and Repository

Perspective: SOA Governance

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Definition

Service Level Definitions > SLD - Ordering application version 2.0

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLD - Ordering application version 2.0

Description

Defines the standard QoS that will be provided by the Ordering application to the Web site

Average Response Time

5 seconds

Availability

24/7 High Availability

Additional Properties

Back

Approve Specification

Revise Specification

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

OrderingApplication

Available Endpoints

None

Bound SCA Export

None

Bound Web Service Port

OrderingApplicationPort

Compatible Service Level Definitions

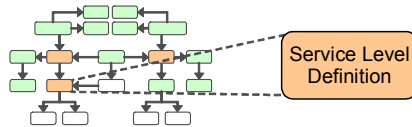
None

Figure 8-5 SOA Governance CoE representative approves the specification of the Ordering application version SLD

This approval moves the Service Level Definition into the Subscribable state, to which consumers can create Service Level Agreement proposals. Additionally, by identifying the service port specifications to which they will adhere, the consumers now have an approved interface and binding against which they can create their consuming application code.

## 8.2.5 Ship Goods Service Level Definition

The Shared Services team now needs to undertake the same steps that the Sporting Goods team has executed in order to define and review their proposed Service Level Definitions.



The Shared Services developer reviews the details of the business QoS requirements, as detailed in the business case and requirements documentation on the service version specification. The

Shared Services developer then uses this information to create a new SLD to define the level of service that will be made available to the Ordering application, as well as to the interface that must be used to interact with the service, as can be seen in Figure 8-6 on page 194.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Definition

Service Versions > Ship goods version 2.1 > SLD - Ship goods version 2.1

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLD - Ship goods version 2.1

Description

Specifies the level of service which will be provided as standard to the Web site

Average Response Time

3.5 seconds

Availability

24/7 High Availability

Additional Properties

Back

Propose Scope

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

ShipGoods

Available Endpoints

None

Bound SCA Export

None

Bound Web Service Port

None

Compatible Service Level Definitions

None

Dependent Entities

Consuming Service Level Agreement(s)

None

Providing Capability Version

Ship goods version 2.1

Governance State

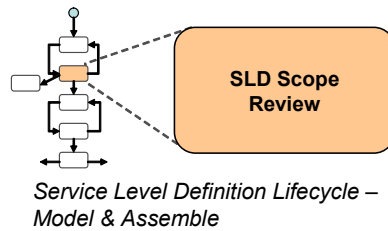
SLD Identified

Classifications

SLD Identified

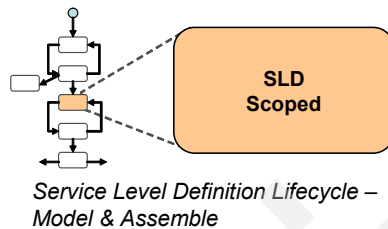
Extended Service Level Definition

Figure 8-6 Shared Services developer creates an SLD for the Ship Goods service version



The developer then proposes this SLD specification for review by the SOA Governance CoE.

## 8.2.6 Ship Goods Service Level Definition scope review



The CoE representative reviews the Service Level Definition that has been specified in order to verify that the proposed qualities of service will meet the business requirements. Additionally, The CoE representative reviews the definition to ensure that it adheres to the JKHLE governance process standards. The CoE representative approves the defined

scope, as shown in Figure 8-7 on page 196.

JKHLE

Service Registry and Repository

Perspective: SOA Governance

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Definition

Service Level Definitions > SLD - Ship goods version 2.1

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLD - Ship goods version 2.1

Description

Specifies the level of service which will be provided as standard to the Web site

Average Response Time

3.5 seconds

Availability

24/7 High Availability

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

ShipGoods

Available Endpoints

None

Bound SCA Export

None

Bound Web Service Port

None

Compatible Service Level Definitions

None

Additional Properties

Back

Approve Scope

Revise Scope

Figure 8-7 SOA Governance CoE representative approves the defined scope of the Ship Goods service version SLD

## 8.2.7 Ship Goods Service Level Definition specification



The developer now elaborates on the SLD specification by identifying the port name and namespace that consumers will use to interact with the application. The developer builds a relationship from the SLD to the service port that was defined in 7.8.1, “Ship Goods endpoint specification” on page 179, as shown in Figure 8-8 on page 197.



JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Edit Relationships

Service Level Definitions > SLD - Ship goods version 2.1 > Edit Relationships

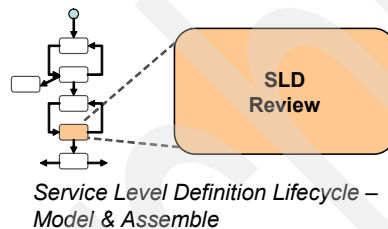
The Extended Service Level Definition 'SLD - Ship goods version 2.1' has the following relationships and targets. Make the required changes and click 'Finish' to apply.

Finish

Cancel

	Targets
<b>SLD - Ship goods version 2.1</b>   <a href="#">Add Relationship</a>	
↳ Available Endpoints	<a href="#">Add Service Endpoint</a>
↳ Compatible Service Level Definitions	<a href="#">Add Service Level Definition</a>
↳ Bound Web Service Port	<a href="#">Add Service Port</a>
	<a href="#">ShipGoodsPort</a>   <a href="#">Replace</a>   <a href="#">Remove</a>
↳ Service Interface	<a href="#">ShipGoods</a>   <a href="#">Replace</a>   <a href="#">Remove</a>
↳ Bound SCA Export	<a href="#">Add Service Export</a>

Figure 8-8 Shared Services developer relates the SLD to the specified endpoints



After completing the specification, the developer can propose this specification for review by the CoE, as shown in Figure 8-9 on page 198.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Definition

Service Level Definitions > SLD - Ship goods version 2.1

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLD - Ship goods version 2.1

Description

Specifies the level of service which will be provided as standard to the Web site

Average Response Time

3.5 seconds

Availability

24/7 High Availability

Additional Properties

Back

Propose Specification

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

ShipGoods

Available Endpoints

None

Bound SCA Export

None

Bound Web Service Port

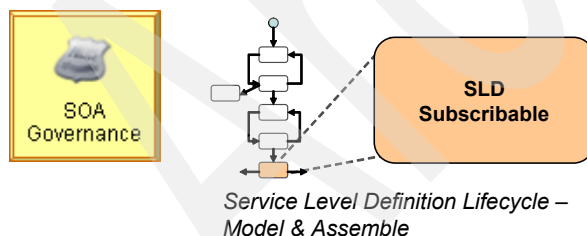
ShipGoodsPort

Compatible Service Level Definitions

None

Figure 8-9 Shared Services developer proposes the specification of the SLD for review

## 8.2.8 Ship Goods Service Level Definition review



The CoE representative reviews the service port specification that has been proposed. The CoE representative verifies that the specified endpoints will provide the bindings suitable for the service consumers and that the JKHLE guidelines are being followed, specifically around the security aspects of the specified protocol. The CoE representative approves the specification, as shown in Figure 8-10 on page 199.

JKHLE

Service Registry and Repository

Perspective: SOA Governance

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Definition

Service Level Definitions > SLD - Ship goods version 2.1

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLD - Ship goods version 2.1

Description

Specifies the level of service which will be provided as standard to the Web site

Average Response Time

3.5 seconds

Availability

24/7 High Availability

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

ShipGoods

Available Endpoints

None

Bound SCA Export

None

Bound Web Service Port

ShipGoodsPort

Compatible Service Level Definitions

None

Additional Properties

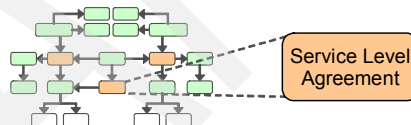
Back

Approve Specification

Revise Specification

Figure 8-10 SOA Governance CoE representative approves the specification of the Ship Goods application version SLD

## 8.3 Creating Service Level Agreements



The Sporting Goods development team instigates the process of defining the technical relationship between the Ordering application and the three provider services: Check Customer Status, Execute

Payment service, and Ship Goods service. The Sporting Goods development lead opens the documents of understanding (DOUs) in WSRR, which were previously approved, and clicks the button to create a new Service Level Agreement (SLA), as highlighted in Figure 8-11 on page 200.

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

DOU

DOUs > DoU Ordering application -> Ship goods

Detail view for DOU. A DOU or Document of Understanding represents an agreement between two Organizations whereby they agree how the consuming Organizations realization of their capability (Capability Version) will use the providing organizations realized capability.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

DoU Ordering application -> Ship goods

Description

DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

Asset Web Link

<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B31C44DE8-2BD0AA3924632%7D&v=2.0>

Remote State

Approved

Owner Email

Requirements Link

<http://reqs.jkhle.com/req-928>

Additional Properties

Back

Deprecate

New SLA

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Consumer

Ordering application version 2.0

Provider

Ship goods version 2.1

Subscribed Service Level Agreements

None

Owning Organization

Shared services

Aggregation

None

Dependency

None

Artifacts

None

Governance State

Asset Approved

Figure 8-11 Sporting Goods development lead creates a new SLA for each of the consumed service versions

The Service Level Agreement (SLA) is the technical implementation of the DOU that allows the context and actual interaction patterns (Service Level Definition, interface, binding, port, endpoint policies, and so forth) to be selected and potentially refined in terms of this particular context. A DOU can actually cover a number of SLAs in multiple contexts or multiple interaction patterns.

When creating the SLA, the development lead defines the quality of service (QoS) and resource requirements that the Ordering application will need in order to satisfy consumer demand, as can be seen in Figure 8-12 on page 201. The

development lead also defines the Service Level Definition that is identified as providing the capabilities necessary to deliver the specified demand.

The screenshot displays the 'Service Registry and Repository' interface. The breadcrumb trail is: DOUs > DoU Ordering application -> Ship goods > SLA - DoU Ordering application -> Ship goods. The page title is 'Extended Service Level Agreement'. Below the title, there's a description: 'Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.' The interface has tabs for 'Details', 'Impact Analysis', 'Governance', 'Policy', and 'Activity'. The 'Details' tab is active, showing various sections: 'Properties', 'Links', 'Relationships', 'Dependent Entities', 'Governance State', and 'Classifications'. The 'Properties' section includes fields for Name, Description, Context Identifier, Service Level Agreement Availability Date, Service Level Agreement Termination Date, Version Match Criteria, and a group of performance metrics (Average Messages Per Day, Maximum Messages Per Day, Minimum Messages Per Day, Peak Message Rate, Peak Message Rate Daily Time, Peak Message Rate Daily Duration) which are highlighted with a red box. The 'Relationships' section shows 'Agreed Endpoints (Service Level Definitions)' with a link to 'SLD - Ship goods version 2.1', also highlighted with a red box. The 'Dependent Entities' section shows 'Consuming Capability Version' as 'Ordering application version 2.0'. The 'Governance State' section shows 'SLA Identified'. The 'Classifications' section shows 'SLA Identified' and 'Extended Service Level Agreement'. At the bottom, there are 'Back' and 'Request SLA' buttons.

**JKHLE Service Registry and Repository** Perspective: Development ▾

Home Actions ▾ View ▾ Tasks ▾ My Service Registry ▾ Help ▾

### Extended Service Level Agreement

**DOUs > DoU Ordering application -> Ship goods > SLA - DoU Ordering application -> Ship goods**

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details Impact Analysis Governance Policy Activity

[Edit Properties](#) | [Edit Relationships](#) | [Edit Classifications](#)

#### Properties

\*Name  
SLA - DoU Ordering application -> Ship good

Description  
DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

Context Identifier

Service Level Agreement Availability Date  
Tuesday, 18 August 2009

Service Level Agreement Termination Date  
Saturday, 1 September 2012

Version Match Criteria  
LatestCompatibleVersion

Average Messages Per Day  
200

Maximum Messages Per Day  
500

Minimum Messages Per Day  
50

Peak Message Rate  
100

Peak Message Rate Daily Time  
18:00

Peak Message Rate Daily Duration  
60

#### Links

- Graphical View
- Applied Policies
- Applied Policy Attachments

#### Relationships

Agreed Endpoints (Service Level Definitions)  
[SLD - Ship goods version 2.1](#)

Service Level Policies

None

Bound SCA Import

None

#### Dependent Entities

Consuming Capability Version  
[Ordering application version 2.0](#)

#### Governance State

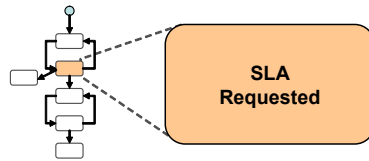
SLA Identified

#### Classifications

SLA Identified  
Extended Service Level Agreement

[Back](#) [Request SLA](#)

Figure 8-12 Sporting Goods development lead defines the quality of service and resourcing required by the Ordering application version



*Service Level Agreement Lifecycle*

The development lead then requests the SLA, as shown in Figure 8-13 on page 203, and notifies the provider development manager that development has documented its requirements.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Agreement

DOUs > DoU Ordering application -> Ship goods > SLA - DoU Ordering application -> Ship goods

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties | Edit Relationships | Edit Classifications

Properties

\*Name

SLA - DoU Ordering application -> Ship good

Description

DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

Context Identifier

Service Level Agreement Availability Date

Tuesday, 18 August 2009

Service Level Agreement Termination Date

Saturday, 1 September 2012

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

100

Peak Message Rate Daily Time

18:00

Peak Message Rate Daily Duration

60

Additional Properties

Back

Request SLA

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Ship goods version 2.1

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Ordering application version 2.0

Governance State

SLA Identified

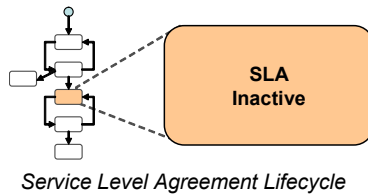
Classifications

SLA Identified

Extended Service Level Agreement

Figure 8-13 Sporting Goods development lead requests the SLA

### 8.3.1 Service Level Agreement approval



The Sporting Goods development lead notifies the provider that there is a proposed SLA for the provider's review. The Shared Services development lead for the Ship Goods service reviews and approves the specified SLA, as shown in Figure 8-14 on page 205.



JKHLE

Service Registry and Repository

Perspective: Development ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Extended Service Level Agreement

Service Level Agreements > SLA - DoU Ordering application -> Ship goods

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLA - DoU Ordering application -> Ship good

Description

DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

Context Identifier

Service Level Agreement Availability Date

Tuesday, 18 August 2009

Service Level Agreement Termination Date

Saturday, 1 September 2012

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

100

Peak Message Rate Daily Time

18:00

Peak Message Rate Daily Duration

60

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Ship goods version 2.1

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Ordering application version 2.0

Governance State

SLA Requested

Classifications

SLA Requested

Extended Service Level Agreement

Additional Properties

Back

Revise SLA Request

Reject SLA Request

Approve SLA Request

Figure 8-14 Shared Services development lead reviews and approves the SLD

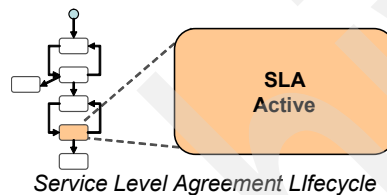
After its approval, this SLA moves into the Inactive state. Therefore, the Sporting Goods development team can continue its development based on the consumption of this specific SLD, but the Sporting Goods development team does not yet have authorization to access any endpoints.

### 8.3.2 Completing the service level specifications



The Sporting Goods organization must also complete these service level agreement steps in order to define the technical specification of the relationship between the Ordering application and the two other (existing) services: Customer Status Check and Execute Payment. The Sporting Goods development lead repeats the steps shown in 8.3, “Creating Service Level Agreements” on page 199. The Shared Service development lead undertakes the steps shown in 8.3.1, “Service Level Agreement approval” on page 204.

### 8.3.3 Service Level Agreement activation



The SLAs for the two existing services (Customer Status Check and Execute Payment) can now be approved for use, because these services do not require any development to support the newly requested SLAs. In order to approve these SLAs for use within the staging environment, the Shared

Services operations lead needs to classify and activate the Service Level Agreements. This activation allows the consumers to successfully invoke either service in the staging environment, therefore allowing the consumers to complete their development and test phases. Figure 8-15 on page 207 shows the Shared Services operations lead activating the SLA for the Execute Payment service, which had previously been classified with the Staging environment classification.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Agreement

Service Level Agreements > SLA - DoU Ordering Application -> Execute Payment

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLA - DoU Ordering Application -> Execute P.

Description

DoU between the Sporting Goods' Ordering Application and the Shared Services' Execute payment

Context Identifier

SGOA020

Service Level Agreement Availability Date

Friday, 7 August 2009

Service Level Agreement Termination Date

Friday, 7 August 2009

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

100

Peak Message Rate Daily Time

18:00

Peak Message Rate Daily Duration

60

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Execute payment

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Ordering application version 2.0

Governance State

SLA Inactive

Classifications

SLA Inactive

Extended Service Level Agreement

Additional Properties

Back

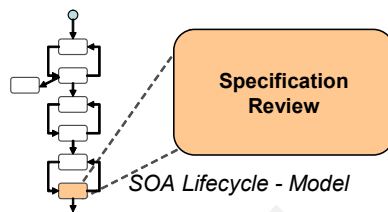
Activate SLA

Figure 8-15 Shared Services operations lead activates the SLA for the Execute Payment service

## 8.4 Version specification approval

The approval of the service level agreements completes the specification of the Ordering application and the Ship Goods service. In order to finalize this specification, the development teams need to propose these specifications for review.

### 8.4.1 Ship Goods specification proposal



The Shared Services developer verifies that all of the necessary steps to finalize the specification have been completed. The Shared Services developer proposes this specification, as shown in Figure 8-16 on page 209.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@jkhle.com

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Specification

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Provider DOU(s)

None

Additional Properties

Back

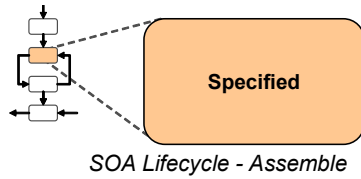
New Version

Propose Specification

New SLD

Figure 8-16 Shared Services developer proposes specification of the Ship Goods service version

## 8.4.2 Ship Goods specification approval



The CoE at this point verifies the specification of the Ship Goods service and all of its related entities (SLD, Interface Specification, and so forth) is complete and in the correct lifecycle states. The CoE also verifies that all JKHLE development and governance processes have been followed correctly. After verifying all of the components of the design, the CoE approves the specification, as shown in Figure 8-17 on page 211.

JKHLE

Service Registry and Repository

Perspective: SOA Governance ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:service:registry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@khle.com

Additional Properties

Back

New Version

Revise Specification

Approve Specification

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Specification

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

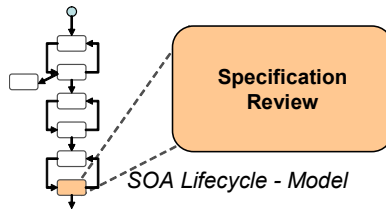
Provider DOU(s)

DoU Ordering application -> Ship goods

Figure 8-17 SOA Governance CoE representative approves the specification of the Ship Goods service version

Chapter 8. Defining Service Level Agreements 211

### 8.4.3 Ordering application specification proposal



The Sporting Goods department now needs to undertake the same specification review and approval step that the Shared Services team undertook for the Ship Goods service.

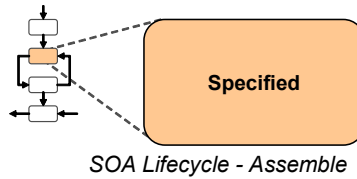
The Sporting Goods developer verifies that all of the steps necessary to finalize the specification are complete and proposes this

specification, as shown in Figure 8-18 on page 213.





#### 8.4.4 Ordering application specification approval



The CoE at this point verifies that the specification of the Ordering application and all of its related entities (SLD, Interface Specification, and so forth) is complete and in the correct lifecycle states. The CoE also verifies that all JKHLE development and governance processes have been followed correctly. After verifying all of the components of the design, the CoE approves the specification, as shown in Figure 8-19 on page 215.

JKHLE

Service Registry and Repository

Perspective: SOA Governance

Home

Actions

View

Tasks

My Service Registry

Help

Application Version

Application Versions > Ordering application version 2.0

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application version 2.0

Description

Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

Version

2.0

Consumer Identifier

SGOA020

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0

Remote State

Approved

Owner Email

debra@jkhle.com

Additional Properties

Back

New Version

Revise Specification

Approve Specification

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ordering application interface specification

Provided Web Services

OrderingApplication-Specification

Provided SCA Modules

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

Ordering application development plan.doc

Provides

SLD - Ordering application version 2.0

Consumes

SLA - DoU Ordering application -> Customer status check

SLA - DoU Ordering application -> Execute payment

SLA - DoU Ordering application -> Ship goods

Dependent Entities

Chartered Business Capability(s)

Ordering Application

Consumer DOU(s)

Figure 8-19 SOA Governance CoE representative approves the specification of the Ordering application version

## 8.5 Progress summary

The two development teams have now completed the specification of the Ordering application and the Ship Goods service. This specification now includes the definition of the levels of service that the deliverables will provide that meet or exceed the requirements requested by the business (shown in Figure 8-20).

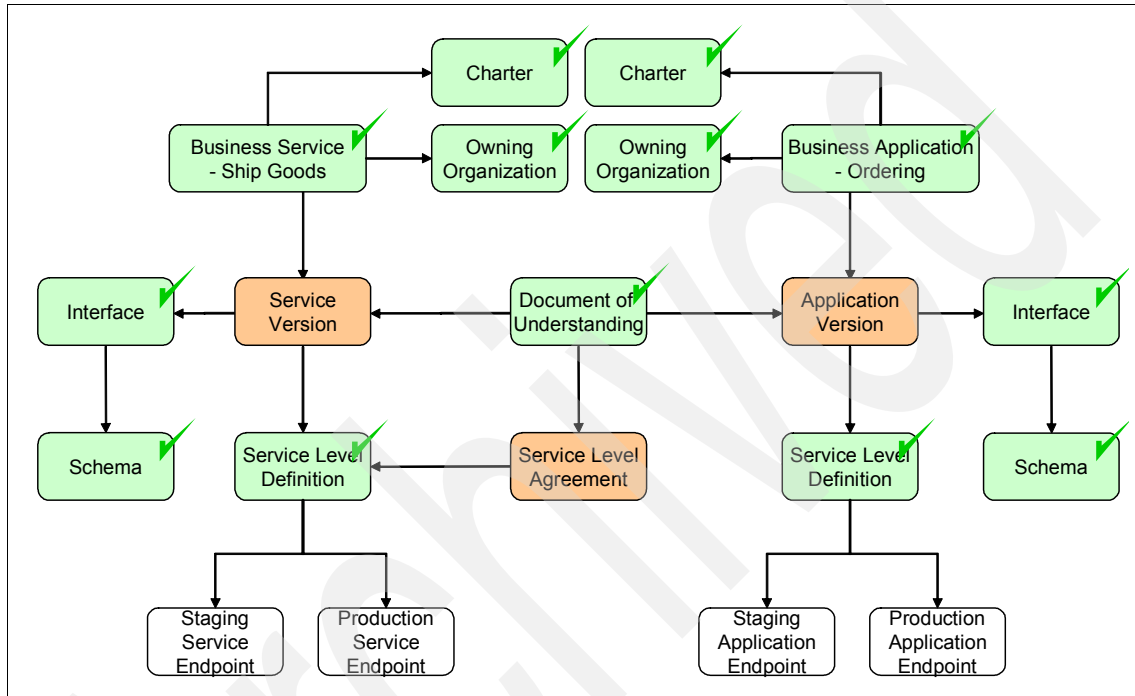


Figure 8-20 Development teams have completed the specifications for the Ordering application and the Ship Goods service

# Release development

In this chapter, we describe the steps that the development teams undertake to move the Ship Goods service into the staging environment and to complete the development of the Ordering application.

This chapter includes these topics:

- ▶ 9.1, “Introduction” on page 218
- ▶ 9.2, “Development implementation” on page 218
- ▶ 9.3, “Ship Goods service realization” on page 218
- ▶ 9.4, “Ordering application realization” on page 233

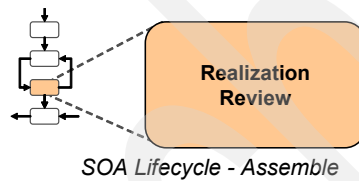
## 9.1 Introduction

The approval of the specification of the service versions and application version allows both the consumers and the providers to proceed with development independently.

## 9.2 Development implementation

The Sporting Goods and Shared Services development teams create the Ordering application and Ship Goods service implementations according to the approved specifications. They undertake unit testing to ensure that the implementations meet the technical specifications. Based on each development cycle, they provide code distribution or code “drops” to the consumer organizations to allow the early verification of capability and interoperability.

## 9.3 Ship Goods service realization



The test teams of the Shared Services organization execute their test plan to verify the implementation capabilities of the Ship Goods service. When these tests have been completed, the test lead identifies an implementation release candidate. Figure 9-1 on page 219 shows the Shared Services test lead proposing the Ship Goods realization.



Service Registry and Repository

Perspective: Development ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@jkhle.com

Additional Properties

Back

New Version

Propose Realization

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Specification

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

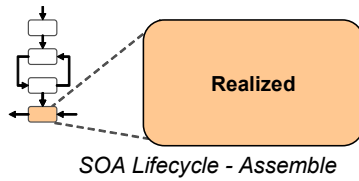
None

Provider DOU(s)

DoU Ordering application -> Ship goods

Figure 9-1 Shared Services test lead proposes a realization of the Ship Goods service version

### 9.3.1 Realization approval



The release manager of the Shared Services organization reviews the proposed realization. After reviewing the status with their development and test teams and confirming that the required development and test processes have been completed, the release manager approves the proposed realization.

Figure 9-2 on page 221 shows the release manager giving this approval.



JKHLE

Service Registry and Repository

Perspective: Development ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@khle.com

Additional Properties

Back

New Version

Revise Realization

Approve Realization

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Specification

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Provider DOU(s)

DoU Ordering application -> Ship goods

Figure 9-2 Shared Services release manager approves the realization of the Ship Goods service version

Chapter 9. Release development 221

## 9.3.2 Staging deployment



The Shared Services operations team lead deploys the proposed build candidate to the staging environment, following the Shared Services operations team's standard deployment procedures to identify the assets to deploy and to undertake the deployment process. After deploying the service and successfully completing the deployment verification tests, the operations lead retrieves the Web Services Description Language (WSDL) for the deployed service and loads this WSDL into WebSphere Service Registry and Repository (WSRR), as shown in Figure 9-3.



Figure 9-3 Shared Services operations lead loads the WSDL containing the staging endpoints for the Ship Goods service version

**Note:** We recommend that the service and schema specification documents are given the same version number as the Capability Version that they implement, when they are loaded into WSRR.

**Note:** The WSDL definition that is being loaded contains this information:

- One WSDLService, which is related to from the service version, in this case, ShipGoodsService-Staging.
- One or more service endpoints per service, which are related to from the SLD that they support, in this case, just the one endpoint:  
`http://staging.jkhle.com:9080/governance/services/sg`

After loading the new WSDL into WSRR, the operations lead needs to classify the new staging endpoint object appropriately for the environment, as shown in Figure 9-4.

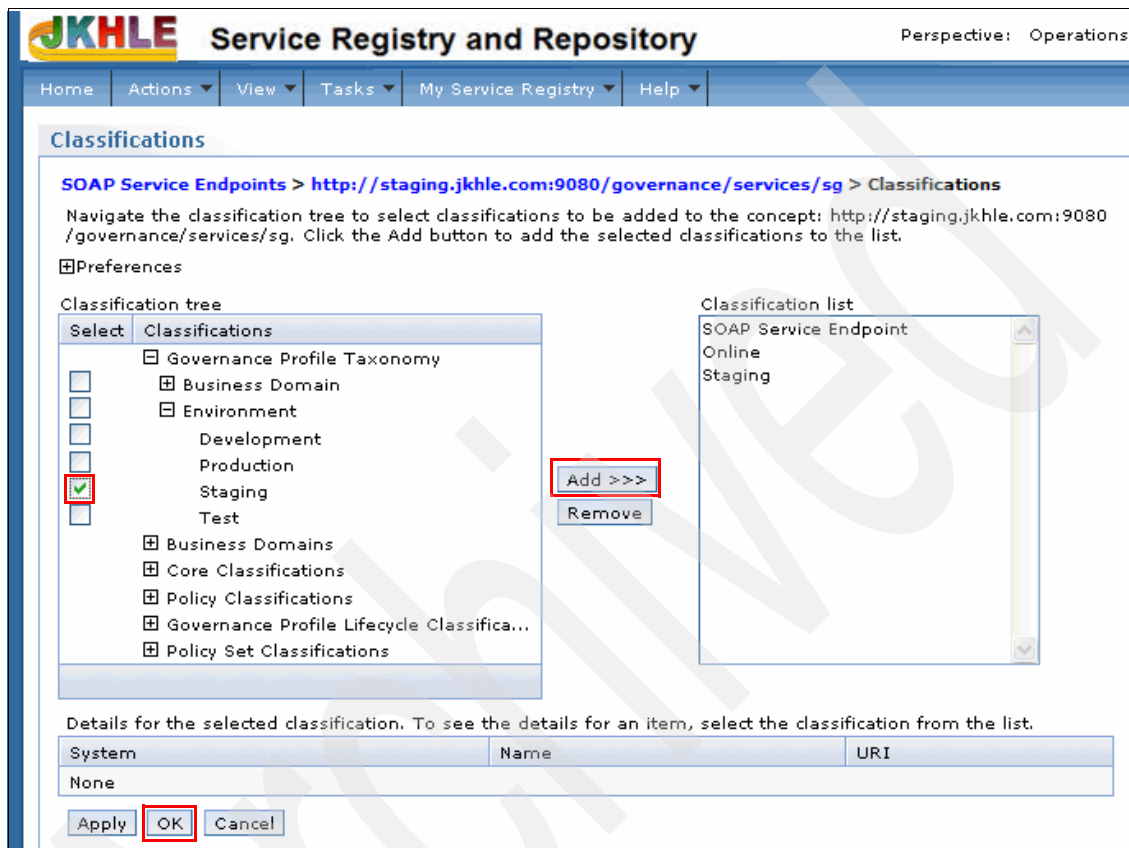


Figure 9-4 Operations lead classifies the endpoint in the Staging environment

With the loading and classification of the service version endpoint complete, the operations lead needs to build the relationship between the Ship Goods service and the service version endpoint that was just loaded and classified, as shown in Figure 9-5 on page 224.

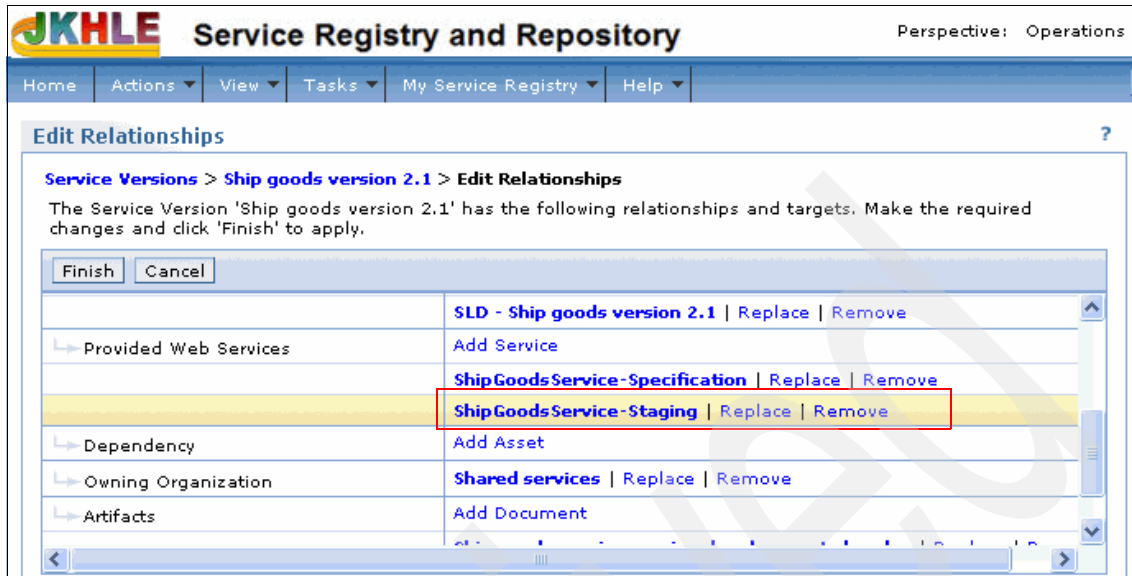
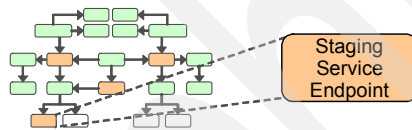


Figure 9-5 Shared Services operations lead relates the staging endpoints with the Ship Goods service version



After loading and classifying the endpoint in WSRR, the operations lead then builds the relationship between the Service Level Definition and the staging endpoint, as shown in Figure 9-6 on page 225.

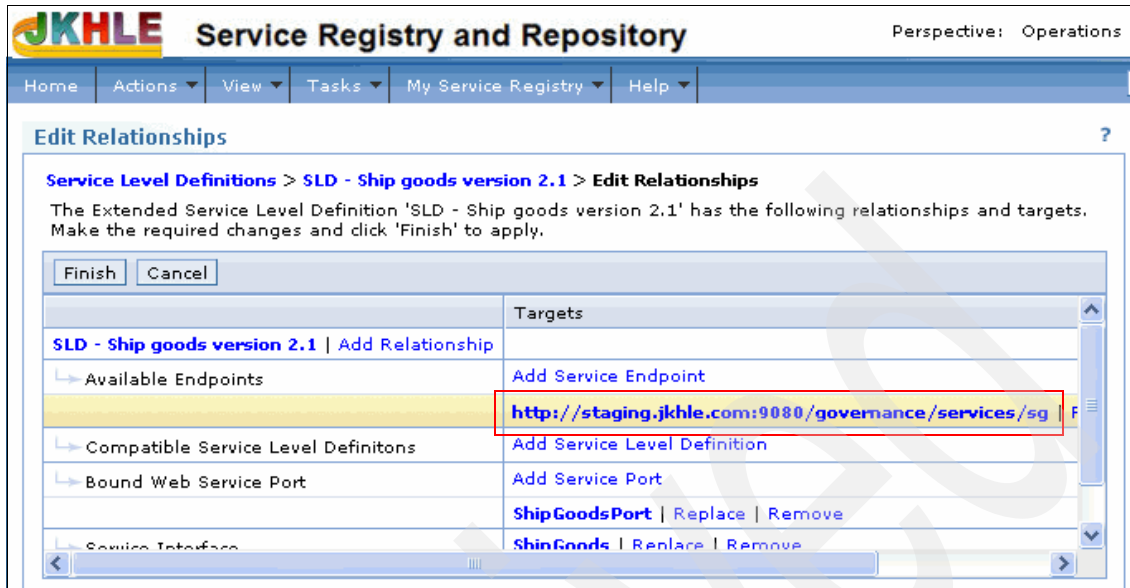
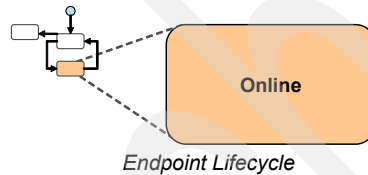


Figure 9-6 Shared Services operations lead relates the staging endpoints to the Service Level Definition



operations lead approves the service endpoint for use to indicate that it is online to the service consumers, as shown in Figure 9-7 on page 226.

The operations lead performs a final verification that the service has been deployed successfully to the staging environment and that it is online. This verification is a check that the service is running, not that it functions correctly. After verifying that the service is available, the

JKHLE

Service Registry and Repository

Perspective: Operations

[Home](#)
[Actions](#)
[View](#)
[Tasks](#)
[My Service Registry](#)
[Help](#)

### SOAP Service Endpoint

[Service Level Definitions](#) > [SLD - Ship goods version 2.1](#) > <http://staging.jkhle.com:9080/governance/services/sg>

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

[Details](#)
[Impact Analysis](#)
[Governance](#)
[Policy](#)
[Activity](#)

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

Properties

\*Name

http://staging.jkhle.com:9080/governance/services/sg

Description

Namespace

http://www.jkhle.com/governance/service/sg

Owner

jon

Version

1.0

Last modified

Tuesday, 4 August 2009 16:56:22 ET

\*Endpoint Type

SOAPAddress

\*Port Name

ShipGoodsPort

Service Version

1.0

\*Service Namespace

http://www.jkhle.com/governance/service/sg

\*Service Name

ShipGoodsService-Staging

Additional Properties

[Back](#)
[Approve For Use](#)
[Retire From Use](#)

Links

[Graphical View](#)
[Applied Policies](#)
[Applied Policy Attachments](#)

Relationships

Soap Address

ShipGoodsPort\_SOAPAddress

WSDL Ports

ShipGoodsPort

Source Document

ShipGoodsStaging.wsdl

Governance State

Offline

Classifications

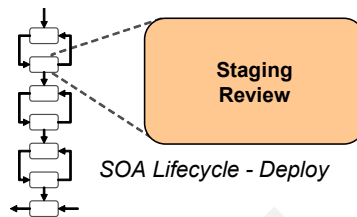
Offline

SOAP Service Endpoint

Staging

Figure 9-7 Shared Services operations lead approves the staging endpoints for use

**Important:** If an enterprise service bus mediation performs an endpoint lookup without enforcing SLA subscriptions or the consumer/provider relationship, it is possible for the bus to invoke this endpoint even though the Capability Version is not in the Staged lifecycle state. The current GEP implementation requires the endpoint in the appropriate environment to be online before the Capability Version is proposed for deployment. If required, this can be changed so that the endpoints were transitioned to being online after the service version has been approved for deployment.



Finally, in order to indicate that the service version has been successfully deployed to the Staging environment, the operations lead proposes the service version, as shown in Figure 9-8 on page 228.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@khle.com

Additional Properties

Back

New Version

Propose Staging Deployment

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Specification

ShipGoodsService-Staging

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Provider DOU(s)

Figure 9-8 Shared Services operations lead proposes the Ship Goods service version staging deployment



### 9.3.3 Staging review

The Shared Services operations and test teams execute a complete set of regression tests, as well as service version-specific test scenarios. These tests verify that the service performs to both the functional and nonfunctional details of the specification, particularly ensuring that the deployed instance meets the requirements of the Service Level Definition. After completing their final verification testing of the service, the Shared Services teams approve the staging deployment, as shown in Figure 9-9 on page 230.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:service:registry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@khle.com

Additional Properties

Back

New Version

Revise Staging Deployment

Approve Staging Deployment

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Specification

ShipGoodsService-Staging

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

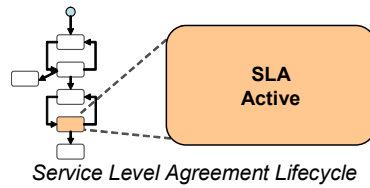
None

Provider DOU(s)

Figure 9-9 Shared Services operations lead approves the staging deployment of the Ship Goods service version

230 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition

### 9.3.4 Service Level Agreement activation



Now that the Ship Goods service has a tested endpoint available for consumers to invoke, the operations lead needs to classify and activate the Service Level Agreement (SLA). This activation allows the consumers to successfully invoke the Ship Goods service in the staging environment, therefore

allowing them to complete their development and test phases. Figure 9-10 on page 232 shows the Shared Services operations lead activating the SLA, which they classify as being available for the Staging environment.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Agreement

Service Level Agreements > SLA - DoU Ordering application -> Ship goods

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLA - DoU Ordering application -> Ship good

Description

DoU between the Sporting Goods' Ordering Application and the Shared Services' Ship Goods

Context Identifier

Service Level Agreement Availability Date

Tuesday, 18 August 2009

Service Level Agreement Termination Date

Saturday, 1 September 2012

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

100

Peak Message Rate Daily Time

18:00

Peak Message Rate Daily Duration

60

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Ship goods version 2.1

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Ordering application version 2.0

Governance State

SLA Inactive

Classifications

SLA Inactive

Extended Service Level Agreement

Staging

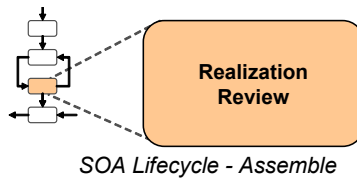
Additional Properties

Back

Activate SLA

Figure 9-10 Shared Services operations lead activates the SLA between the Ordering application version and the Ship Goods service version

## 9.4 Ordering application realization



The Sporting Goods development and test teams use the staging environment endpoints of the three consumed services to execute their various test plans. When these test cycles have been completed, the test lead identifies an implementation release candidate.

Figure 9-11 on page 234 shows the Shared Services test lead proposing the Ordering application realization.

**Service Registry and Repository**

Perspective: Development ▾

[Home](#)
[Actions ▾](#)
[View ▾](#)
[Tasks ▾](#)
[My Service Registry ▾](#)
[Help ▾](#)

### Application Version

**Application Versions > Ordering application version 2.0**

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

[Details](#)
[Impact Analysis](#)
[Governance](#)
[Policy](#)
[Activity](#)

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

**Properties**

- \*Name  
Ordering application version 2.0
- Description  
Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for
- Version  
2.0
- Consumer Identifier  
SGOA020
- Version Availability Date  
Tuesday, 1 September 2009
- Version Termination Date  
Tuesday, 4 August 2009
- Version Requirements Link  
[urn:serviceregistry](#)
- Asset Web Link  
[https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0](#)
- Remote State  
Approved
- Owner Email  
[debra@jkhle.com](#)

**Links**

- [Graphical View](#)
- [Applied Policies](#)
- [Applied Policy Attachments](#)

**Relationships**

- Interface Specifications  
[Ordering application interface specification](#)
- Provided Web Services  
[OrderingApplication-Specification](#)
- Provided SCA Modules  
None
- Owning Organization  
[Sporting goods](#)
- Dependency  
None
- Aggregation  
None
- Artifacts  
[Ordering application development plan.doc](#)
- Provides  
[SLD - Ordering application version 2.0](#)
- Consumes  
[SLA - DoU Ordering application -> Customer status check](#)  
[SLA - DoU Ordering application -> Execute payment](#)  
[SLA - DoU Ordering application -> Ship goods](#)

**Additional Properties**

[Back](#)
[New Version](#)
[Propose Realization](#)

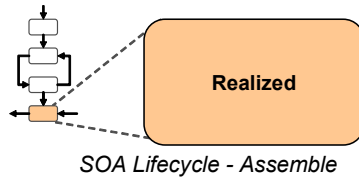
[New SLD](#)

**Dependent Entities**

- Chartered Business Capability(s)  
[Ordering Application](#)
- Consumer DOU(s)

Figure 9-11 Sporting Goods test lead proposes the realization of the Ordering application

### 9.4.1 Realization approval



The release manager of the Sporting Goods organization reviews the proposed realization. After reviewing the status with the development and test teams and having confirmed that the required development and test processes have completed, the release manager approves the proposed realization. Figure 9-12 on page 236 shows the release manager giving this approval.

**Service Registry and Repository**

Perspective: Development ▾

[Home](#)
[Actions ▾](#)
[View ▾](#)
[Tasks ▾](#)
[My Service Registry ▾](#)
[Help ▾](#)

### Application Version

**Application Versions > Ordering application version 2.0**

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

[Details](#)
[Impact Analysis](#)
[Governance](#)
[Policy](#)
[Activity](#)

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

**Properties**

- \*Name: Ordering application version 2.0
- Description: Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for shipping.
- Version: 2.0
- Consumer Identifier: SGOA020
- Version Availability Date: Tuesday, 1 September 2009
- Version Termination Date: Tuesday, 4 August 2009
- Version Requirements Link: urn:service:registry
- Asset Web Link: https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-34B0D0-2CDA-AD17805C0E40%7D&v=2.0
- Remote State: Approved
- Owner Email: debra@jkhle.com

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

- Interface Specifications: Ordering application interface specification
- Provided Web Services: OrderingApplication-Specification
- Provided SCA Modules: None
- Owning Organization: Sporting goods
- Dependency: None
- Aggregation: None
- Artifacts: Ordering application development plan.doc
- Provides: SLD - Ordering application version 2.0
- Consumes:
  - SLA - DoU Ordering application -> Customer status check
  - SLA - DoU Ordering application -> Execute payment
  - SLA - DoU Ordering application -> Ship goods

**Additional Properties**

[Back](#)
[New Version](#)
[Revise Realization](#)

[Approve Realization](#)
[New SLD](#)

**Dependent Entities**

- Chartered Business Capability(s): Ordering Application
- Consumer DOU(s):

Figure 9-12 Sporting Goods release manager approves realization of the Ship Goods service version



## 9.5 Progress summary

Figure 9-13 shows that the Shared Services development team has completed the development of the Ship Goods service and has completed the deployment to the Staging environment. The Shared Services operations team has classified the SLA appropriately and made it available for consumption within the Staging environment, by the Ordering application.

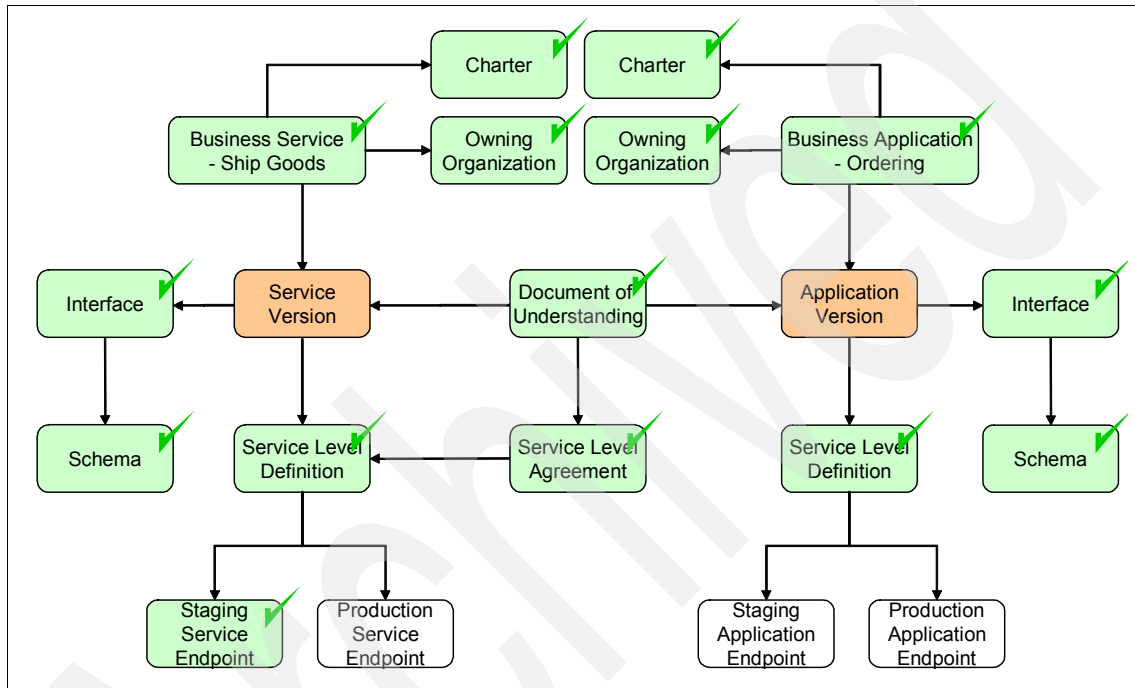


Figure 9-13 Ship Goods service has been deployed to the staging environment



## Application staging

This chapter examines the steps that are necessary to move the Ordering application into the staging environment. It then looks at the process that is required to test both the Ordering application and the Ship Goods service. The chapter also looks at the process that is required for the consumers to certify the implementation adherence to the specifications.

This chapter includes these topics:

- ▶ 10.1, “Ordering application staging deployment” on page 240
- ▶ 10.2, “Ship Goods consumer verification” on page 246
- ▶ 10.3, “Ordering application consumer verification” on page 250
- ▶ 10.4, “Application and service certification” on page 252

## 10.1 Ordering application staging deployment



With the development and testing of the Ordering application complete, the Sporting Goods operations team lead deploys the proposed build candidate to the staging environment. After deploying the service and successfully completing the deployment verification tests, the operations lead retrieves the Web Services Description Language (WSDL) for the deployed service and loads this WSDL into WebSphere Service Registry and Repository (WSRR), as shown in Figure 10-1.



Figure 10-1 Sporting Goods operations lead loads the WSDL containing the staging endpoints

The operations lead classifies the new staging endpoint object appropriately for the environment.

After loading and classifying the application version endpoint, the operations lead then builds the relationship between the Ordering application and the version endpoint as shown in Figure 10-2 on page 241.

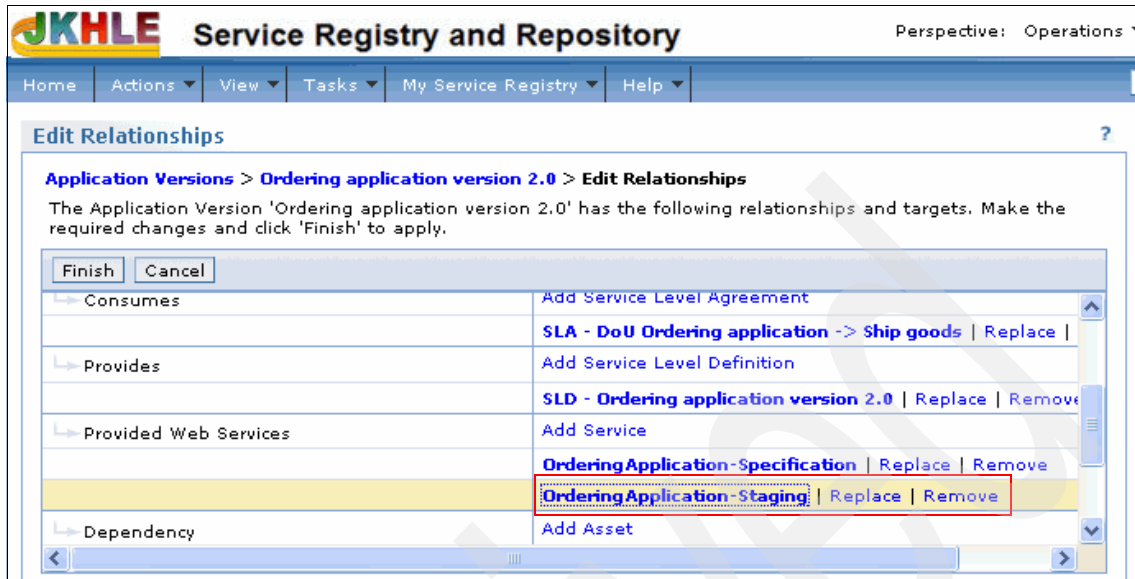
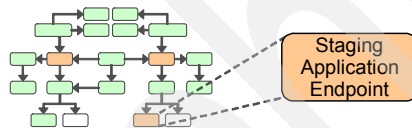


Figure 10-2 Sporting Goods operations lead relates the Ordering application version with the staging endpoints



The operations lead then builds the relationship between the Service Level Definition (SLD) and the staging endpoint, as shown in Figure 10-3 on page 242.

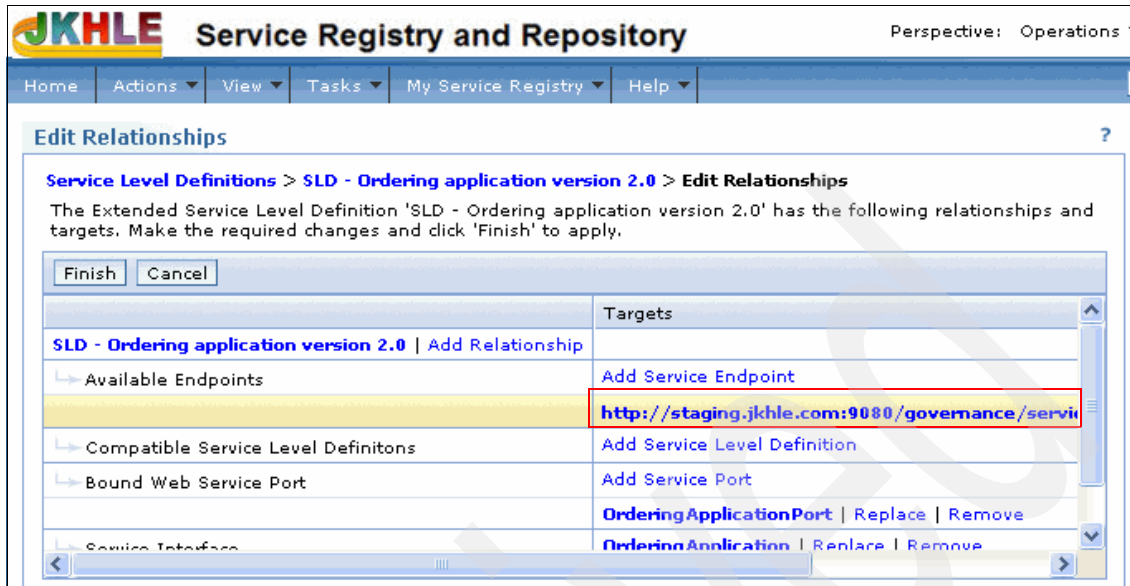
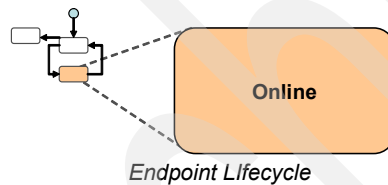


Figure 10-3 Sporting Goods operations lead relates the SLD with the staging endpoints



The operations lead then performs a final verification that the application has been deployed successfully to the staging environment and that it is online. This verification is a check that the application is running, not that it functions correctly. After verifying that the application is available, the operations lead approves the application for use to indicate that it is online to the application consumers, as shown in Figure 10-4 on page 243.



Service Registry and Repository

Perspective: Operations

[Home](#)
[Actions](#)
[View](#)
[Tasks](#)
[My Service Registry](#)
[Help](#)

### SOAP Service Endpoint

[Service Level Definitions](#) > 
 [SLD - Ordering application version 2.0](#) > 
 <http://staging.jkhle.com:9080/governance/services/oa>

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

[Details](#)
[Impact Analysis](#)
[Governance](#)
[Policy](#)
[Activity](#)

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

Properties

\*Name

http://staging.jkhle.com:9080/governance/services/oa

Description

Namespace

http://www.jkhle.com/governance/service/oa

Owner

lisa

Version

1.0

Last modified

Tuesday, 4 August 2009 17:13:11 ET

\*Endpoint Type

SOAPAddress

\*Port Name

OrderingApplicationPort

Service Version

1.0

\*Service Namespace

http://www.jkhle.com/governance/service/oa

\*Service Name

OrderingApplication-Staging

Additional Properties

Back

Approve For Use

Retire From Use

Links

[Graphical View](#)
[Applied Policies](#)
[Applied Policy Attachments](#)

Relationships

Soap Address

OrderingApplicationPort\_SOAPAddress

WSDL Ports

OrderingApplicationPort

Source Document

OrderingApplicationStaging.wsdl

Governance State

Offline

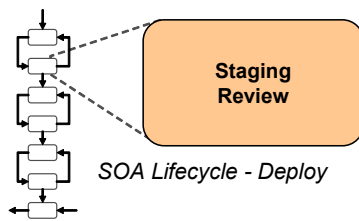
Classifications

Offline

SOAP Service Endpoint

Staging

Figure 10-4 Sporting Goods operations lead approves the staging endpoint for use



Finally, in order to indicate that the application version has been successfully deployed to the Staging environment, the operations lead proposes the application version, as shown in Figure 10-5 on page 245.



JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Application Version

Application Versions > Ordering application version 2.0

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application version 2.0

Description

Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

Version

2.0

Consumer Identifier

SGOA020

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0

Remote State

Approved

Owner Email

debra@jkhle.com

Additional Properties

Back

New Version

Propose Staging Deployment

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ordering application interface specification

Provided Web Services

OrderingApplication-Specification

OrderingApplication-Staging

Provided SCA Modules

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

Ordering application development plan.doc

Provides

SLD - Ordering application version 2.0

Consumes

SLA - DoU Ordering application -> Ship goods

Dependent Entities

Chartered Business Capability(s)

Ordering Application

Consumer DOU(s)

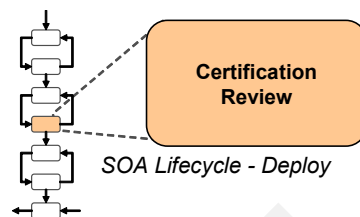
DoU Ordering Application -> Customer Status Check

DoU Ordering Application -> Execute Payment

Figure 10-5 Sporting Goods operations lead proposes the Ordering application version for staging deployment

## 10.2 Ship Goods consumer verification

With both the Ordering application and the Ship Goods service in the staging environment, the Sporting Goods operations and test teams execute a set of test plans to verify both the functional and nonfunctional capabilities of the deployed entities. These tests certify that the Ship Goods service and the Ordering application meet all of the requirements to be ready for deployment, including verification of service and application compliance with their SLDs and Service Level Agreements (SLAs).



Upon completion of the test plans, the Sporting Goods operations lead first confirms that the Ship Goods service version meets the Sporting Goods operations lead's requirements by proposing the service certification, as shown in Figure 10-6 on page 247.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@khle.com

Additional Properties

Back

New Version

Propose Certification

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Specification

ShipGoodsService-Staging

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

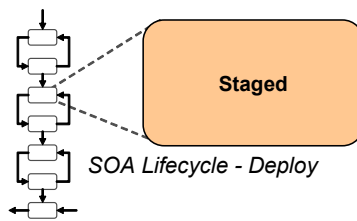
Ship goods

Consumer DOU(s)

None

Provider DOU(s)

Figure 10-6 Sporting Goods operations lead proposes Ship Goods service version for certification



The Sporting Goods operations lead then provides the approval of the Ordering application version staging deployment, as shown in Figure 10-7 on page 249.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Application Version

Application Versions > Ordering application version 2.0

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application version 2.0

Description

Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

Version

2.0

Consumer Identifier

SGOA020

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0

Remote State

Approved

Owner Email

debra@jkhle.com

Additional Properties

Back

New Version

Revise Staging Deployment

Approve Staging Deployment

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ordering application interface specification

Provided Web Services

OrderingApplication-Specification

OrderingApplication-Staging

Provided SCA Modules

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

Ordering application development plan.doc

Provides

SLD - Ordering application version 2.0

Consumes

SLA - DoU Ordering application -> Ship goods

Dependent Entities

Chartered Business Capability(s)

Ordering Application

Consumer DOU(s)

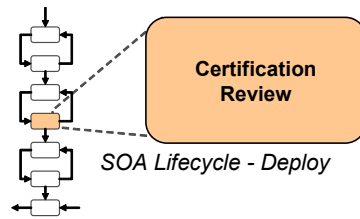
DoU Ordering Application -> Customer Status Check

DoU Ordering Application -> Execute Payment

Figure 10-7 Sporting Goods operations lead approves staging deployment of Ordering application version

This approval of the Ordering application staging deployment means that the consumers of this new application can undertake their consumer acceptance testing.

## 10.3 Ordering application consumer verification



The Sporting Goods development, test, and operations teams certify that the Ordering application version satisfies the specification requirements and that it is ready for deployment to production. To make this application version available for certification by the Center of Excellence (CoE), the operations lead proposes the certification, as shown in Figure 10-8 on page 251.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Application Version

Application Versions > Ordering application version 2.0

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application version 2.0

Description

Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

Version

2.0

Consumer Identifier

SGOA020

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3B0D0-2CDA-AD17805C0E40%7D&v=2.0

Remote State

Approved

Owner Email

debra@jkhle.com

Additional Properties

Back

New Version

Propose Certification

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ordering application interface specification

Provided Web Services

OrderingApplication-Specification

OrderingApplication-Staging

Provided SCA Modules

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

Ordering application development plan.doc

Provides

SLD - Ordering application version 2.0

Consumes

SLA - DoU Ordering application -> Ship goods

Dependent Entities

Chartered Business Capability(s)

Ordering Application

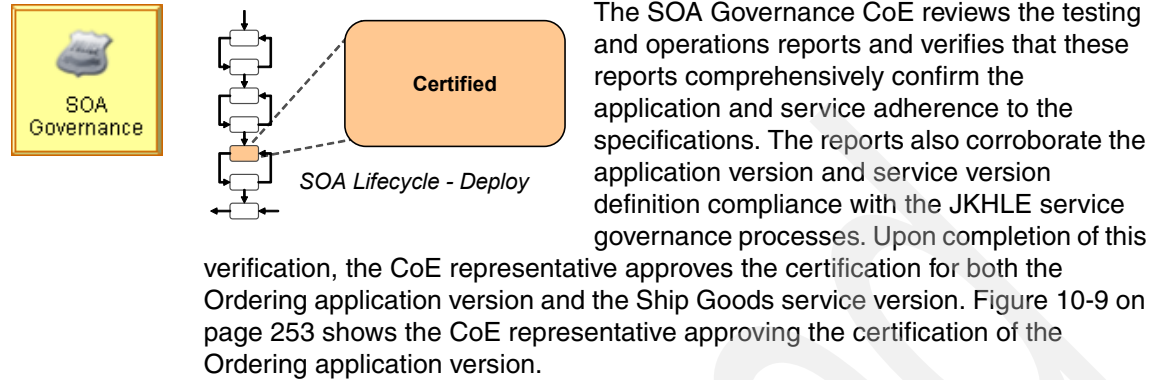
Consumer DOU(s)

DoU Ordering Application -> Customer Status Check

Figure 10-8 Sporting Goods operations lead proposes Ordering application version for certification

Chapter 10. Application staging 251

## 10.4 Application and service certification





JKHLE

Service Registry and Repository

Perspective: SOA Governance ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Application Version

Application Versions > Ordering application version 2.0

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ordering application version 2.0

Description

Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

Version

2.0

Consumer Identifier

SGOA020

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-3<B0D0-2CDA-AD17805C0E40%7D&v=2.0

Remote State

Approved

Owner Email

debra@jkhle.com

Additional Properties

Back

New Version

Revise Certification

Approve Certification

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ordering application interface specification

Provided Web Services

OrderingApplication-Specification

OrderingApplication-Staging

Provided SCA Modules

None

Owning Organization

Sporting goods

Dependency

None

Aggregation

None

Artifacts

Ordering application development plan.doc

Provides

SLD - Ordering application version 2.0

Consumes

SLA - DoU Ordering application -> Ship goods

Dependent Entities

Chartered Business Capability(s)

Ordering Application

Consumer DOU(s)

DoU Ordering Application -> Customer Status Check

DoU Ordering Application -> Execute Payment

Figure 10-9 SOA Governance CoE representative approves certification of the Ordering application version

Chapter 10. Application staging 253

## 10.5 Progress summary

Figure 10-10 shows that the development teams have now completed the development and testing of the Ordering application and the Ship Goods service. Both the Ordering application and the Ship Goods service are now successfully deployed to and verified in the Staging environment, and they have been certified for deployment to production.

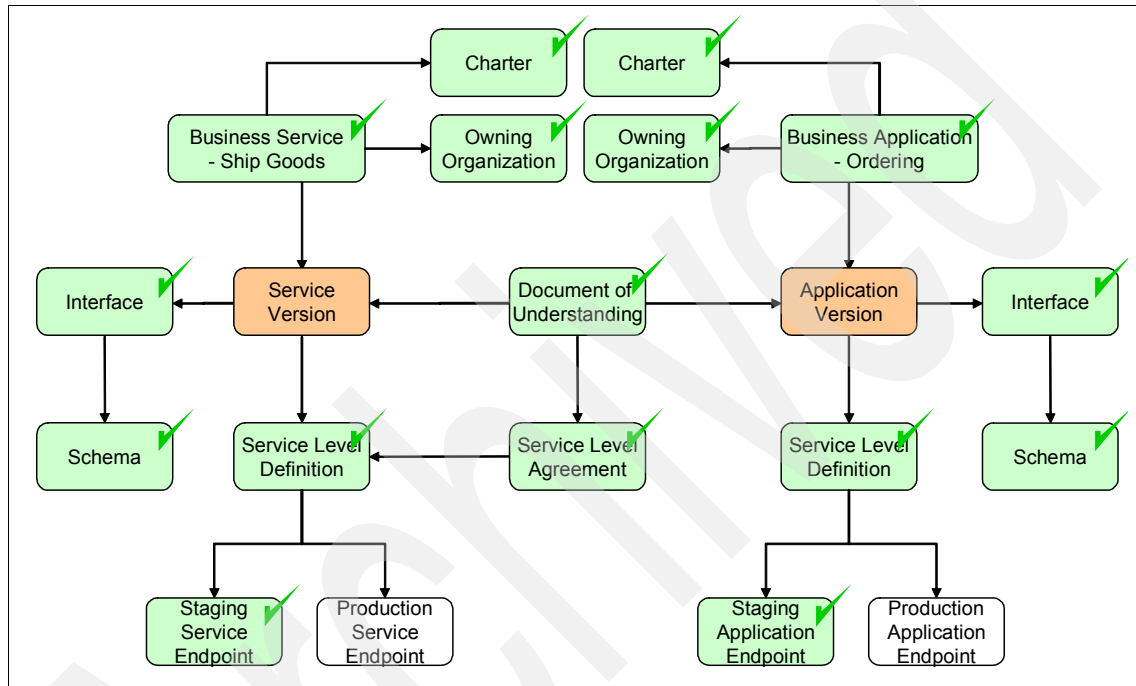


Figure 10-10 Ordering application and Ship Goods service have been deployed to and verified in the staging environment

# Production deployment

The final chapter in this part describes the service governance steps that are required to deploy the Ship Goods and Ordering application releases to production.

This chapter discusses these topics:

- ▶ 11.1, “Ship Goods service production deployment” on page 256
- ▶ 11.2, “Ordering application production deployment” on page 264

## 11.1 Ship Goods service production deployment



The Ship Goods service version is now ready to be deployed to production. The Shared Services operations team lead deploys the proposed build candidate to the production environment, following the standard deployment procedures to identify the assets to deploy and to undertake the deployment process. After deploying the service version and after successfully completing the deployment verification tests, the operations team lead retrieves the Web Services Description Language (WSDL) for the deployed service version and loads this WSDL into WSRR, as shown in Figure 11-1.



Figure 11-1 Shared Services operations team lead loads the WSDL containing the production endpoints

The operations team lead classifies the new production endpoint object appropriately for the environment.

With the new service version endpoint loaded and classified, the operations team lead builds the relationship between the Ship Goods service version and the service version endpoint as shown in Figure 11-2 on page 257.

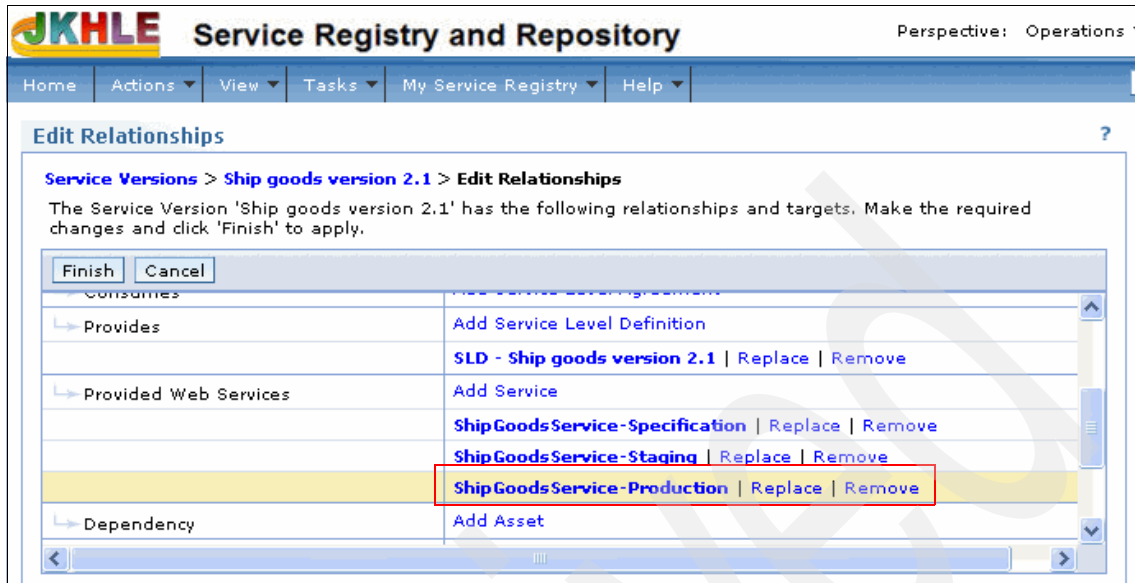
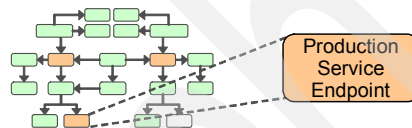


Figure 11-2 Shared Services operations team lead relates the Ship Goods service version with the production endpoints



The operations team lead builds the relationship between the Service Level Definition and the production endpoint, as shown in Figure 11-3 on page 258.

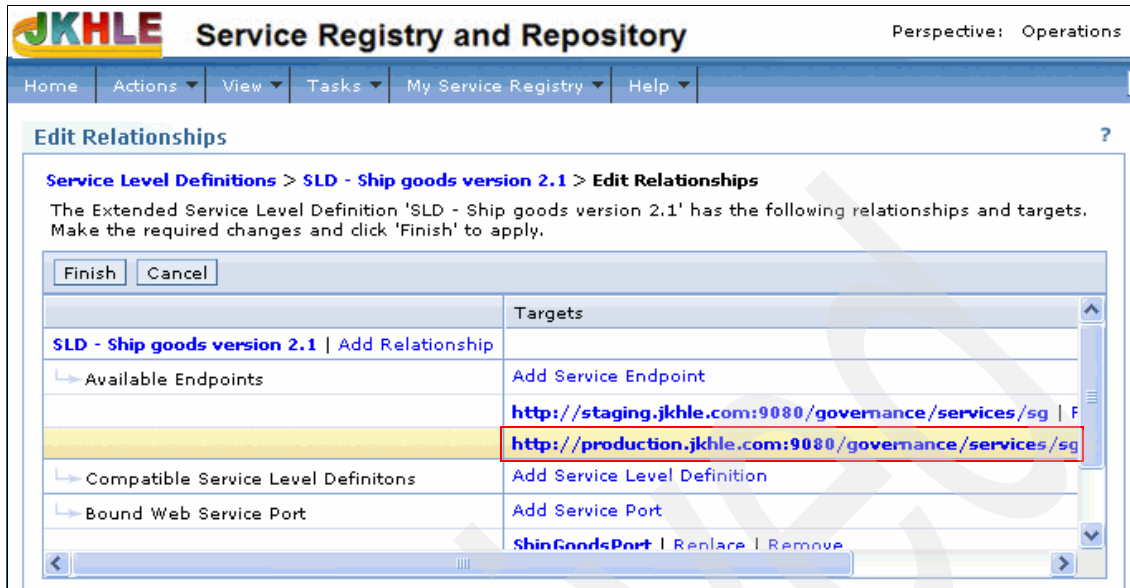
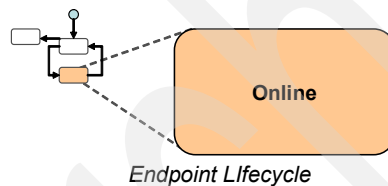


Figure 11-3 Shared Services operations team lead relates the Service Level Definition with the production endpoints



The operations team lead performs a final verification that the service has been deployed successfully to the production environment and that it is online. This verification is a check that the service is running, not that it functions correctly. After verifying that the service is available, the operations team lead approves the service endpoint for use to indicate that it is online to the service consumers, as shown in Figure 11-4 on page 259.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

SOAP Service Endpoint

Service Level Definitions > SLD - Ship goods version 2.1 > http://production.jkhle.com:9080/governance/services/sg

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

http://production.jkhle.com:9080/governance/services/sg

Description

Namespace

http://www.jkhle.com/governance/service/sg

Owner

jon

Version

1.0

Last modified

Tuesday, 4 August 2009 17:24:46 ET

\*Endpoint Type

SOAPAddress

\*Port Name

ShipGoodsPort

Service Version

1.0

\*Service Namespace

http://www.jkhle.com/governance/service/sg

\*Service Name

ShipGoodsService-Production

Additional Properties

Back

Approve For Use

Retire From Use

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Soap Address

ShipGoodsPort\_SOAPAddress

WSDL Ports

ShipGoodsPort

Source Document

ShipGoodsProduction.wsdl

Governance State

Offline

Classifications

Offline

Production

SOAP Service Endpoint

Figure 11-4 Shared Services operations team lead approves the production endpoint for use

In order to make the service level agreement (SLA) active in the production environment, the Shared Services operations team lead classifies the SLA in the Production environment, as shown in Figure 11-5 on page 260.

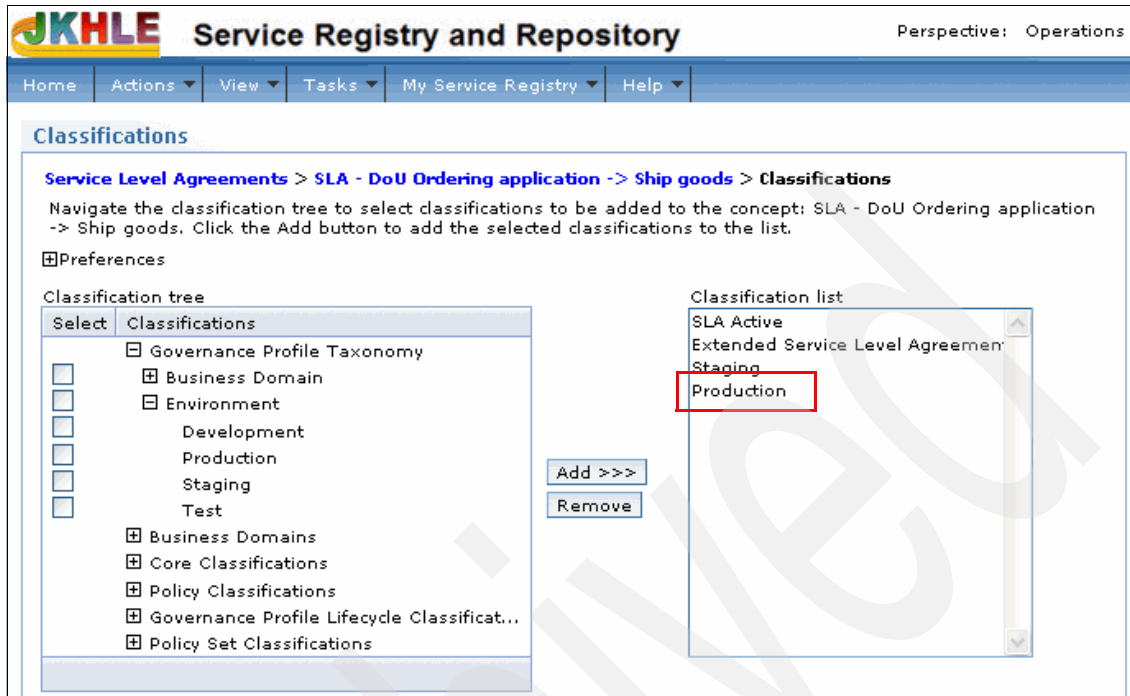
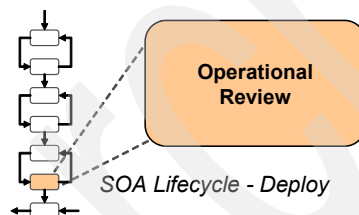


Figure 11-5 Shared Services operations team lead classifies the SLA in Production



Finally, in order to indicate that the service version has been successfully deployed to the production environment, the operations team lead proposes the service version, as shown in Figure 11-6 on page 261.



JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@khle.com

Additional Properties

Back

New Version

Propose Production Deployment

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Production

ShipGoodsService-Specification

ShipGoodsService-Staging

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

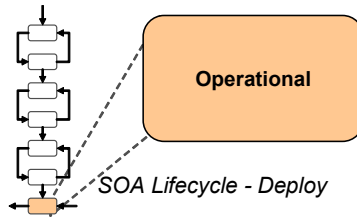
Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Figure 11-6 Shared Services operations team lead proposes the Ship Goods service version for production deployment



At this point, the service is in the production environment, but it is not officially released for production use. This final go-ahead responsibility belongs to the business sponsors of the implementation. These business sponsors liaise with the service-oriented architecture (SOA) Governance Center of Excellence (CoE) and development teams to get the confirmation of acceptance that the service is ready. The business analyst provides confirmation that the line of business (LoB) accepts the service version as available for consumption, as shown in Figure 11-7 on page 263.

JKHLE

Service Registry and Repository

Perspective: Business

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Versions > Ship goods version 2.1

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods version 2.1

Description

Send the order to the warehouse for processing and shipment to the customer anywhere within the USA or Canada

Version

2.1

Consumer Identifier

SSSG021

Version Availability Date

Tuesday, 1 September 2009

Version Termination Date

Tuesday, 4 August 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7BF246660C-9B4C-705E-A965-447E9BE157D6%7D&v=2.1

Remote State

Approved

Owner Email

connie@khle.com

Additional Properties

Back

New Version

Revise Production Deployment

Approve Production Deployment

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

Ship Goods interface specification

Provided Web Services

ShipGoodsService-Production

ShipGoodsService-Specification

ShipGoodsService-Staging

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

Ship goods service version development plan.doc

Provides

SLD - Ship goods version 2.1

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Provider DOU(s)

Figure 11-7 Business analyst approves the Ship Goods Service version for production deployment

Chapter 11. Production deployment 263

The service version is now fully deployed and operational, and the development cycle is complete. This service version's lifecycle can only now be modified to deprecate and ultimately retire this version of the service when it is either superseded or no longer required.

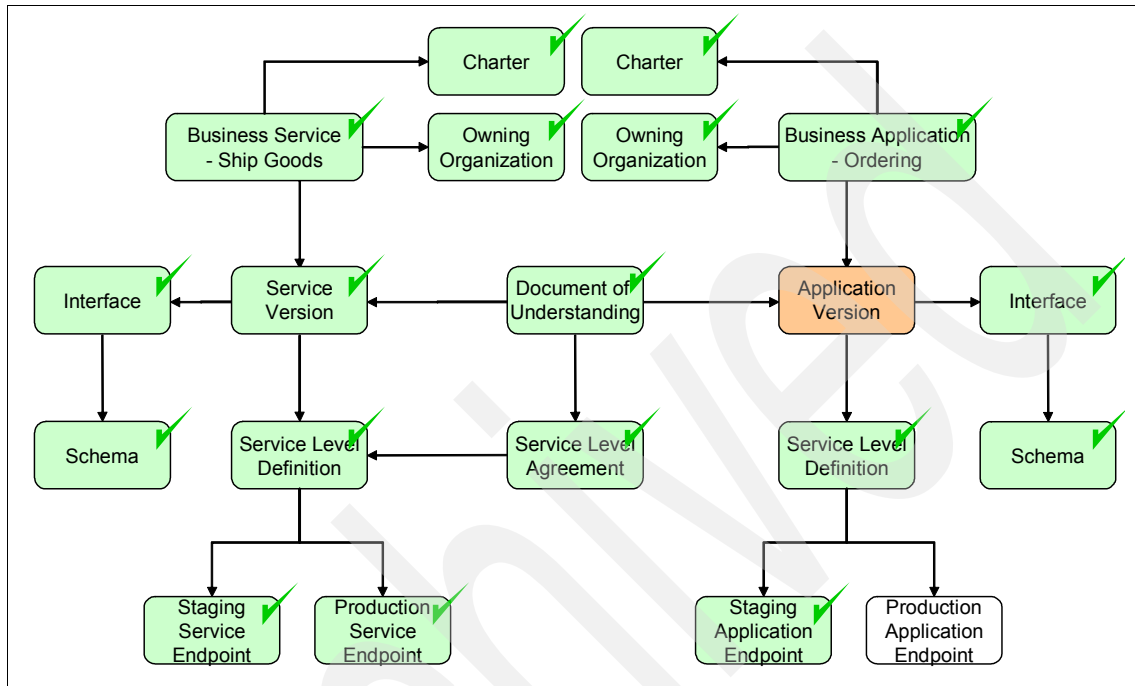


Figure 11-8 Business model structure for the Ship Goods service version

## 11.2 Ordering application production deployment

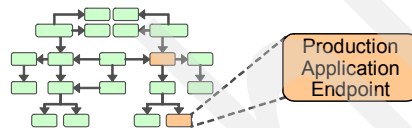


The Ordering application version is now ready to be deployed to production. The Sporting Goods operations team lead deploys the proposed build candidate to the production environment, following JKHLE's standard deployment procedures to identify the assets to deploy and to undertake the deployment process. After deploying the application and successfully completing the deployment verification tests, the operations team lead retrieves the WSDL for the deployed application and loads this WSDL into WSRR, as shown in Figure 11-9 on page 265.



Figure 11-9 Sporting Goods operations team lead loads the WSDL containing the production endpoints

The Sporting Goods operations team lead classifies the new production endpoint object appropriately for the environment.



After loading and classifying the application version endpoint, the Sporting Goods operations team lead needs to build the relationship between the Ordering application and the application endpoint as shown in Figure 11-10 on page 266.

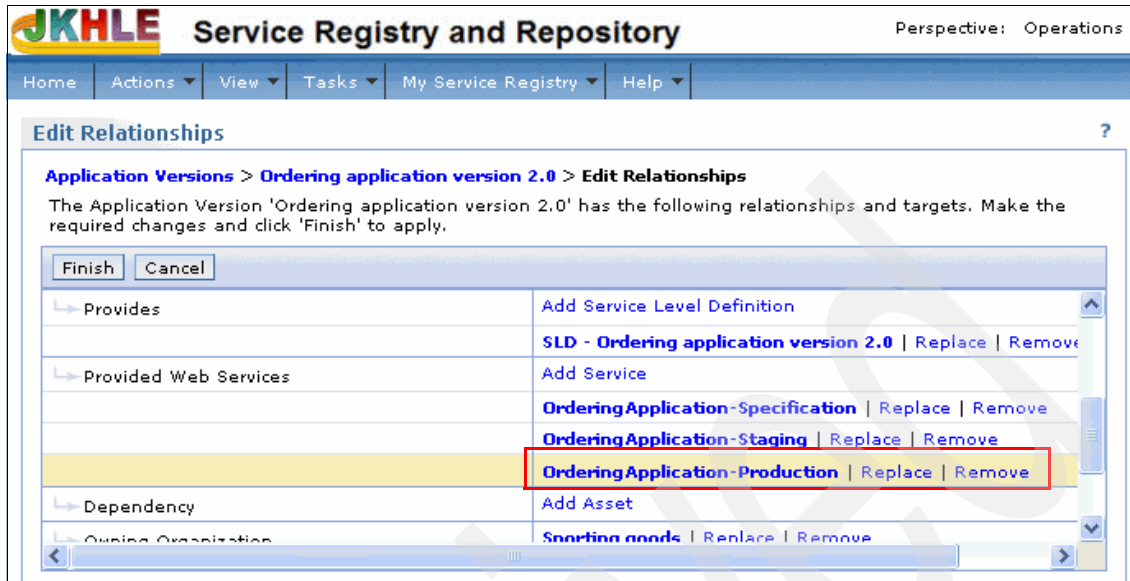


Figure 11-10 Sporting Goods operations team lead relates the Ordering application version with the production endpoints

The Sporting Goods operations team lead builds the relationship between the Service Level Definition (SLD) and the production endpoint, as shown in Figure 11-11 on page 267.

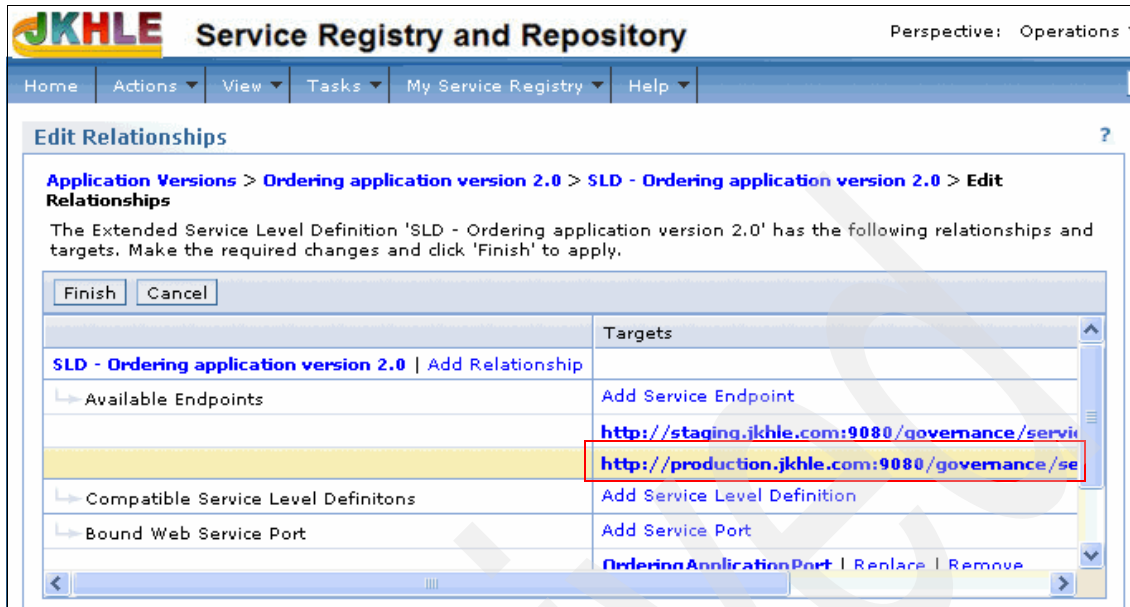
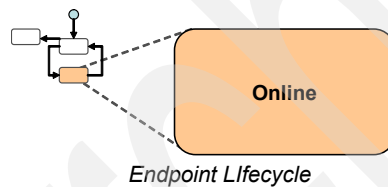


Figure 11-11 Sporting Goods operations team lead relates the Service Level Definition with the production endpoints



The Sporting Goods operations lead performs a final verification that the application has been deployed successfully to the production environment and that it is online. This verification is a check that the application is running, not that it functions correctly. After verifying that the application is available, the operations team lead approves the endpoint for use to indicate that it is online to the consumers, as shown in Figure 11-12 on page 268.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

SOAP Service Endpoint

Application Versions > Ordering application version 2.0 > SLD - Ordering application version 2.0 > http://production.jkhle.com:9080/governance/services/oa

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

http://production.jkhle.com:9080/governance/services/oa

Description

Namespace

http://www.jkhle.com/governance/service/oa

Owner

lisa

Version

1.0

Last modified

Tuesday, 4 August 2009 17:35:55 ET

\*Endpoint Type

SOAPAddress

\*Port Name

OrderingApplicationPort

Service Version

1.0

\*Service Namespace

http://www.jkhle.com/governance/service/oa

\*Service Name

OrderingApplication-Production

Additional Properties

Back

Approve For Use

Retire From Use

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Soap Address

OrderingApplicationPort\_SOAPAddress

WSDL Ports

OrderingApplicationPort

Source Document

OrderingApplicationProduction.wsdl

Governance State

Offline

Classifications

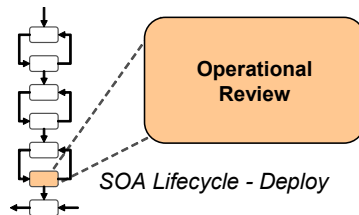
Offline

Production

SOAP Service Endpoint

Figure 11-12 Sporting Goods operations team lead approves the production endpoint for use





Finally, in order to indicate that the application version has been successfully deployed to the production environment, the Sporting Goods operations team lead proposes the application version, as shown in Figure 11-13 on page 269.

**JKHLE Service Registry and Repository** Perspective: Operations | lisa | Logo

Home | Actions | View | Tasks | My Service Registry | Help

### Application Version

**Application Versions > Ordering application version 2.0**

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details | Impact Analysis | Governance | Policy | Activity

[Edit Properties](#) | [Edit Relationships](#) | [Edit Classifications](#)

#### Properties

**\*Name**  
Ordering application version 2.0

**Description**  
Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for

**Version**  
2.0

**Consumer Identifier**  
SGOA020

**Version Availability Date**  
Tuesday, 1 September 2009

**Version Termination Date**  
Tuesday, 4 August 2009

**Version Requirements Link**  
[urn:serviceregistry](#)

**Asset Web Link**  
<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-34B0D0-2CDA-AD17805C0E40%7D&w=2.0>

**Remote State**  
Approved

**Owner Email**  
debra@jkhle.com

#### Links

- Graphical View
- Applied Policies
- Applied Policy Attachments

#### Relationships

**Interface Specifications**  
[Ordering application interface specification](#)

**Provided Web Services**  
[OrderingApplication-Production](#)  
[OrderingApplication-Specification](#)  
[OrderingApplication-Staging](#)

**Provided SCA Modules**  
None

**Owning Organization**  
[Sporting goods](#)

**Dependency**  
None

**Aggregation**  
None

**Artifacts**  
[Ordering application development plan.doc](#)

**Provides**  
[SLD - Ordering application version 2.0](#)

**Consumes**  
[SLA - DoU Ordering application -> Ship goods](#)

#### Dependent Entities

**Chartered Business Capability(s)**  
[Ordering Application](#)

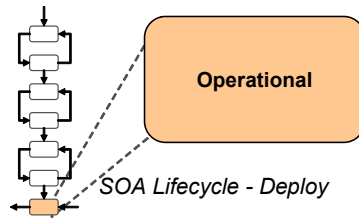
**Consumer DOU(s)**  
[DoU Ordering Application -> Customer Status](#)

**Additional Properties**

[Back](#) [New Version](#) [Propose Production Deployment](#) [New SLD](#)

**Field help**  
For field help information, a field label or field marker when a cursor appears

Figure 11-13 Sporting Goods operations team lead proposes the Ordering application version for production deployment



At this point, the application is in the production environment, but it is not officially released for production use. This final go-ahead responsibility belongs to the business sponsors of the implementation, who liaise with the SOA Governance CoE and development teams to get the confirmation of acceptance that the application is ready. The business analyst provides confirmation that the LoB accepts the application version as available for consumption, as shown in Figure 11-14 on page 271.

Detail view for Application Version. An Application Version represents a specific version (or release) of a Web Application. It identifies the services it consumes with Service Level Agreements to the particular Service Level Definitions provided. It should not provide any services as it is simply a consumer of services.

Details	Impact Analysis	Governance	Policy	Activity
<a href="#">Edit Properties</a>   <a href="#">Edit Relationships</a>   <a href="#">Edit Classifications</a>				
<b>Properties</b>				
<p><b>*Name</b></p> <p>Ordering application version 2.0</p> <p><b>Description</b></p> <p>Process a sporting goods order by checking the customer status, take payment from the customer and send the order to the warehouse for</p> <p><b>Version</b></p> <p>2.0</p> <p><b>Consumer Identifier</b></p> <p>SGOA020</p> <p><b>Version Availability Date</b></p> <p>Tuesday, 1 September 2009</p> <p><b>Version Termination Date</b></p> <p>Tuesday, 4 August 2009</p> <p><b>Version Requirements Link</b></p> <p>urn:serviceregistry</p> <p><b>Asset Web Link</b></p> <p>https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B78F680A3-34B0D0-2CDA-AD17805C0E40%7D&amp;v=2.0</p> <p><b>Remote State</b></p> <p>Approved</p> <p><b>Owner Email</b></p> <p>debra@jkhle.com</p>				
<b>Additional Properties</b>				
<p><a href="#">Back</a> <a href="#">New Version</a></p> <p><a href="#">Revise Production Deployment</a></p> <p><a href="#">Approve Production Deployment</a> <a href="#">New SLD</a></p>				
<b>Links</b>				
<ul style="list-style-type: none"> <li><a href="#">Graphical View</a></li> <li><a href="#">Applied Policies</a></li> <li><a href="#">Applied Policy Attachments</a></li> </ul>				
<b>Relationships</b>				
<p><b>Interface Specifications</b></p> <p><a href="#">Ordering application interface specification</a></p> <p><b>Provided Web Services</b></p> <p><a href="#">OrderingApplication-Production</a></p> <p><a href="#">OrderingApplication-Specification</a></p> <p><a href="#">OrderingApplication-Staging</a></p> <p><b>Provided SCA Modules</b></p> <p>None</p> <p><b>Owning Organization</b></p> <p><a href="#">Sporting goods</a></p> <p><b>Dependency</b></p> <p>None</p> <p><b>Aggregation</b></p> <p>None</p> <p><b>Artifacts</b></p> <p><a href="#">Ordering application development plan.doc</a></p> <p><b>Provides</b></p> <p><a href="#">SLD - Ordering application version 2.0</a></p> <p><b>Consumes</b></p> <p><a href="#">SLA - DoU Ordering application -&gt; Ship goods</a></p>				
<b>Dependent Entities</b>				
<p><b>Chartered Business Capability(s)</b></p> <p><a href="#">Ordering Application</a></p> <p><b>Consumer DOU(s)</b></p> <p><a href="#">DoU Ordering Application -&gt; Customer Status Check</a></p> <p><a href="#">DoU Ordering Application -&gt; Execute Payment</a></p>				

Figure 11-14 Business analyst approves the Ordering application version for production deployment

## 11.3 Progress summary

The application version is now fully deployed and operational, and the development cycle is complete. This application version's lifecycle can only now be modified to deprecate and ultimately retire this version of the application when it is either superseded or no longer required.

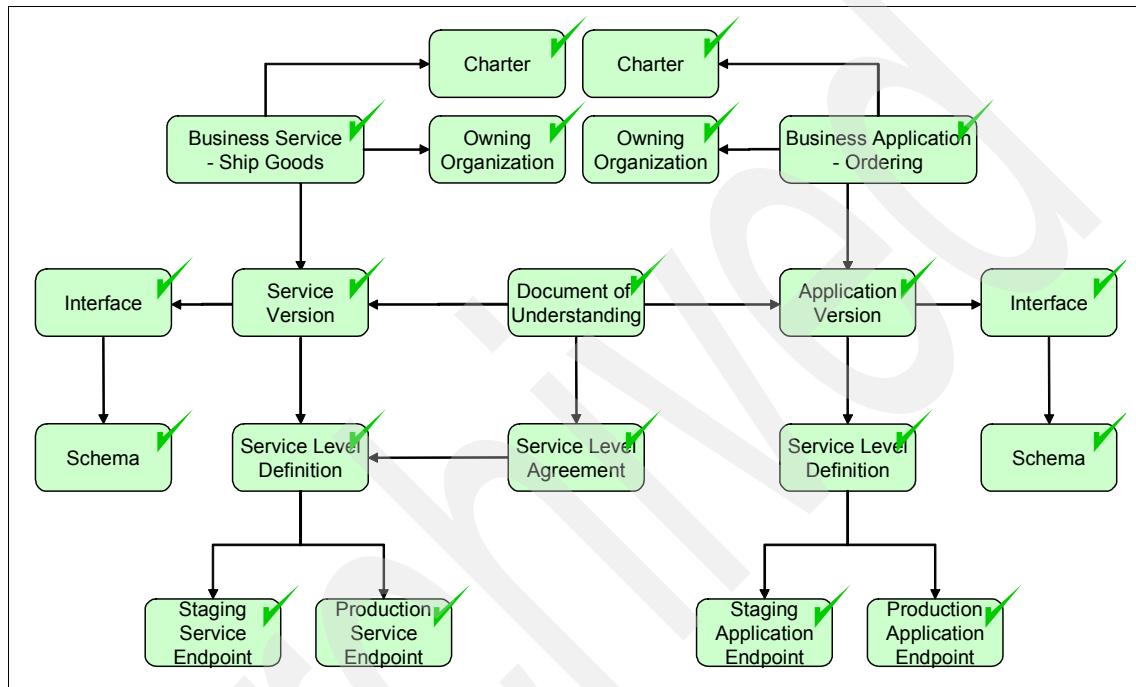


Figure 11-15 Business model structure for the Ordering application version

This step concludes the service governance process, which the JKHL Enterprises' lines of business executed in order to deliver the new Ordering application business capability to the Sporting Goods organization. The process has examined the steps that the various roles within the business units executed in order to deliver the required capabilities.



# Part 3

## Other scenarios

In this part of the book, we look at several additional scenarios that were described in Chapter 3, “Scenario overview” on page 73. Each chapter in this section addresses one of the scenarios that was described. Not all of the scenarios are explicitly described, because they are implicitly covered in detail in other sections of this book.



## Bottom-up scenario

This chapter looks at how the WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition enables JKHL Enterprises to expose an existing service that is already in production by using a bottom-up approach.

This chapter describes the following topics:

- ▶ 12.1, “Introduction” on page 276
- ▶ 12.2, “Registering the Eligibility service WSDL” on page 276
- ▶ 12.3, “Defining the Business Capability for the Eligibility service” on page 278
- ▶ 12.4, “Providing retrospective scope and planning information for the Eligibility service” on page 281
- ▶ 12.5, “Enabling governed consumption and reuse” on page 284
- ▶ 12.6, “Completing the SOA service lifecycle for the Eligibility service” on page 286

# 12.1 Introduction

In this scenario, the Shared Services development team has discovered a service that is in production, but it is not registered within WSRR Advanced Lifecycle Edition. No governance and control protect the service, because it is not cataloged. This service was identified in the Services Overview table while using IBM Tivoli Composite Application Manager for SOA; a service operation was observed in the run time, which had not been registered in WSRR. The service identified is JKHL Enterprises' Eligibility service, which identifies whether a customer satisfies the commercial credentials to be an account holder.

The following sections of this chapter identify the process that is required within WSRR Advanced Lifecycle Edition to make the Eligibility service visible.

# 12.2 Registering the Eligibility service WSDL

The Eligibility service is currently deployed to the production environment and is used by development, test/staging, and other lines of business. The service is defined by a monolithic Web Services Description Language (WSDL) that defines the service, endpoint, binding, portType, and message types. The WSDL was obtained from the deployed service using the `?wsdl` command against the endpoint.

The first step in registering this new service is to load it into WSRR and associate it with a new service version that will be used to govern it.

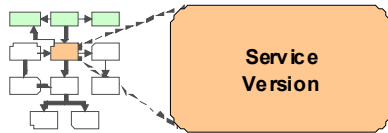


The development team, with guidance from the Service-Oriented Architecture (SOA) Governance Center of Excellence (CoE), loads the service definition for the Eligibility service into WSRR. Figure 12-1 shows that the Eligibility service was loaded into WSRR by the development team.



Figure 12-1 Eligibility service WSDL has been loaded by the development team





A service version representation of the technical Eligibility service WSDL must now be created in WSRR so that complete and valid metadata can be specified describing how the service works in production. We

describe the elaboration details for this new service version representation of the Eligibility service in 7.3, “Shipment service scope and sizing” on page 140. This discovered service already conforms to the JKHL Enterprises’ governance processes and be complementary to their existing services.

Figure 12-2 on page 278 shows the details specified for the new service version:

- ▶ The owning organization is Shared Services.
- ▶ The artifact relevant to this service is the WSDL document.
- ▶ The provided Web services relationship can be set to the correlated Service entity that was created when the WSDL document was loaded.
- ▶ The consumer identifier that is used for this service is SSES040.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Messages

The object named "Eligibility Service" of type "Service Version" was successfully created.

Service Versions > Eligibility Service

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Eligibility Service

Description

This service identifies whether a customer satisfies the commercial credentials to be an account holder.

Version

4.0.0

Consumer Identifier

SSES040

Version Availability Date

Monday, 1 June 2009

Version Termination Date

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

EligibilityService

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Figure 12-2 New Eligibility service version details page

## 12.3 Defining the Business Capability for the Eligibility service

An important part of scoping any service is to identify how it adds value to the business. WSRR uses business capabilities to define the business view of a service. JKHL Enterprises does not already have an existing Business Capability that the Eligibility service realizes so we must now create and define a business service capability. The development lead, in conjunction with the business unit leader for Shared Services, exposes this new Business Capability.

278 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition



The development lead must add these key defining components:

- ▶ A service charter to define the scope and role of the business service
- ▶ An owning organization, which is responsible for defining the requirements for the capability and for owning any realizations of this capability, such as the Eligibility service that is being loaded

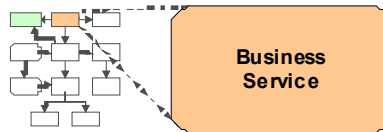


Figure 12-3 shows the new business service that has been defined for the Eligibility service.

**JKHLE Service Registry and Repository** Perspective: Development

Home Actions View Tasks My Service Registry Help

### Business Service

☐ Messages

The object named "Account eligibility" of type "Business Service" was successfully created.

**Business Services > Account eligibility**

Detail view for Business Service. A Business Service represents a business capability that is viewed as a service within the organization. Business Services have realizations (Service Versions) that are implemented using SOA technology and conform to Service Interface Specifications.

Details Impact Analysis Governance Policy Activity

[Edit Properties](#) | [Edit Relationships](#) | [Edit Classifications](#)

☐ **Properties**

\*Name  
Account eligibility

Description  
Validate customer information is complete and can be verified before opening an account.

Business Requirements Link  
[urn:serviceregistry](#)

Asset Web Link  
[urn:serviceregistry](#)

Remote State

Owner Email

☐ **Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

☐ **Relationships**

Charter  
[AccountEligibilityCharter.doc](#)

Service Interface Versions  
None

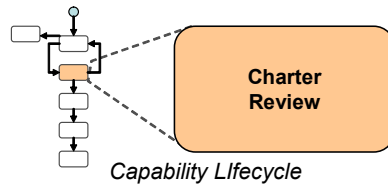
Versions  
None

Owning Organization  
[Shared services](#)

Dependency  
None

Artifacts

Figure 12-3 The business service representing a valuable business capability has been created



This business service capability must now be proposed for charter review by the business analyst, who provided the charter documentation that the development team added to the Business Capability. This step is beneficial, because the business analyst is made aware of the existing service available in JKHL Enterprises as early as possible.



Now that the business service capability has been created and proposed for charter review in WSRR, synchronization will occur, which creates a matching business solution in Rational Asset Manager. This Business Capability is now visible to the SOA Governance Center of Excellence (CoE) team to initiate the approval process. The appropriate approval reviews occur, eventually placing the Business Capability into the Business Capability Approved state in WSRR.

Figure 12-4 on page 281 shows the business analyst or CoE team connecting the newly approved Business Capability to the service version, which was created in 12.2, “Registering the Eligibility service WSDL” on page 276, by modifying the Versions relationship on the Account eligibility business service capability.

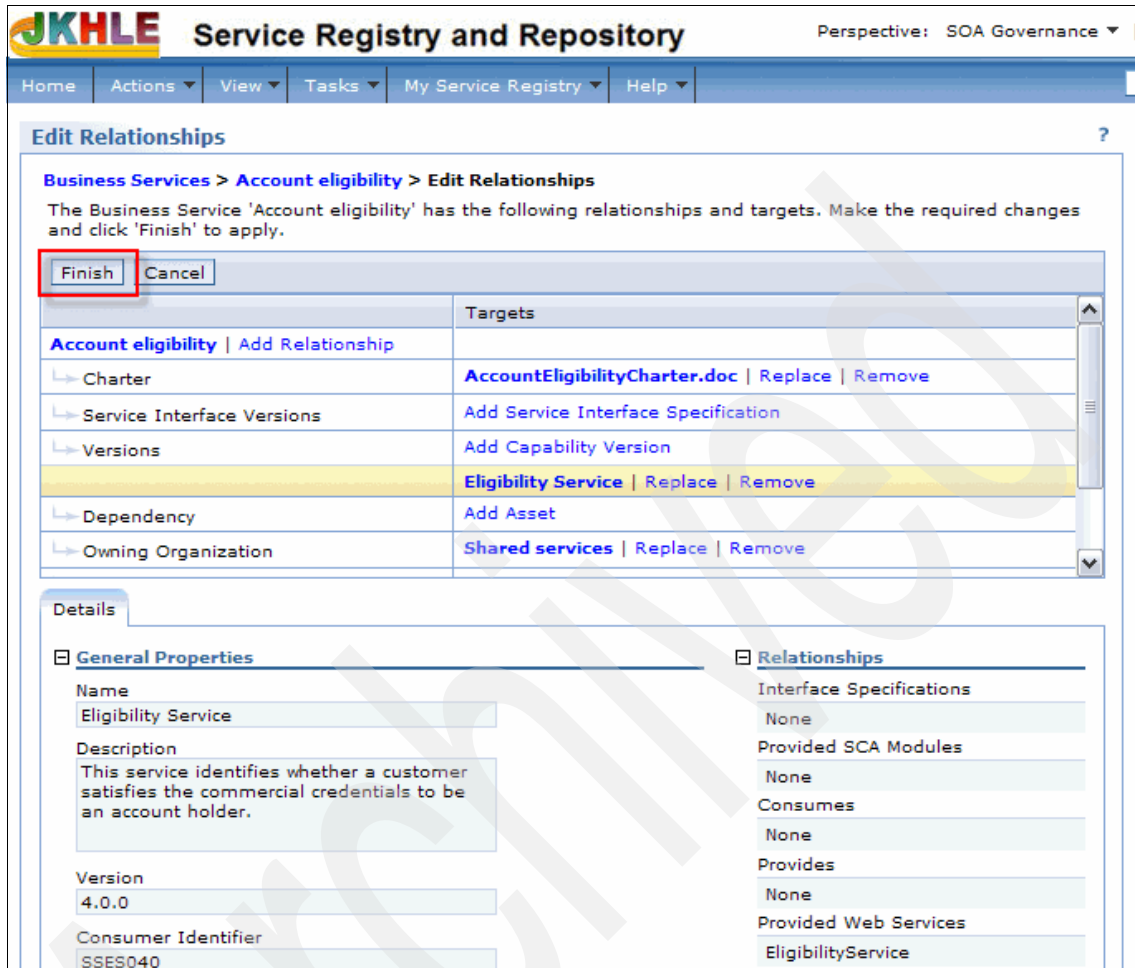


Figure 12-4 Linking the Business Capability to the Eligibility Service service version

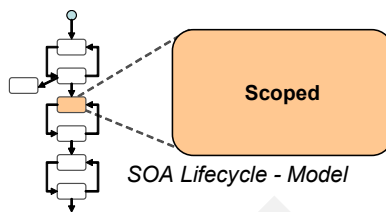
## 12.4 Providing retrospective scope and planning information for the Eligibility service

The business service capability has been linked to the service version of the discovered production Eligibility service. The Eligibility service is already in production, but the metadata associated with the service version with respect to scoping and development planning (as described during the top-down scenario in 7.3, “Shipment service scope and sizing” on page 140) is missing. This service version needs to pass through the Scope and Plan lifecycle states for this

relevant information to be added by the correct teams. The information includes the service capabilities and a likely date to decommission the service (its termination date).



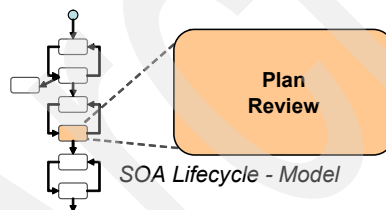
The Shared Services development team amends the Eligibility service version by specifying the updated properties on the Asset, as well as possibly attaching additional documentation to the artifacts relationship before proposing it for Scope Review.



The SOA Governance CoE team analyzes this updated information to ensure that the documentation and details are representative of the production service before approving it as Scoped.



At this stage, the Shared Services development team must specify when the service is likely to be decommissioned (the service planning information) by updating the termination date property on the service version before passing the service on to the Shared Services Line of Business (LoB) manager or business unit leader.



The Shared Services business unit leader initiates the review process in Rational Asset Manager.

Here, the release manager and the business unit leader have the opportunity to add comments, review, and accept the retrospective planning information. Figure 12-5 shows the final approval step for the service version in Rational Asset Manager.

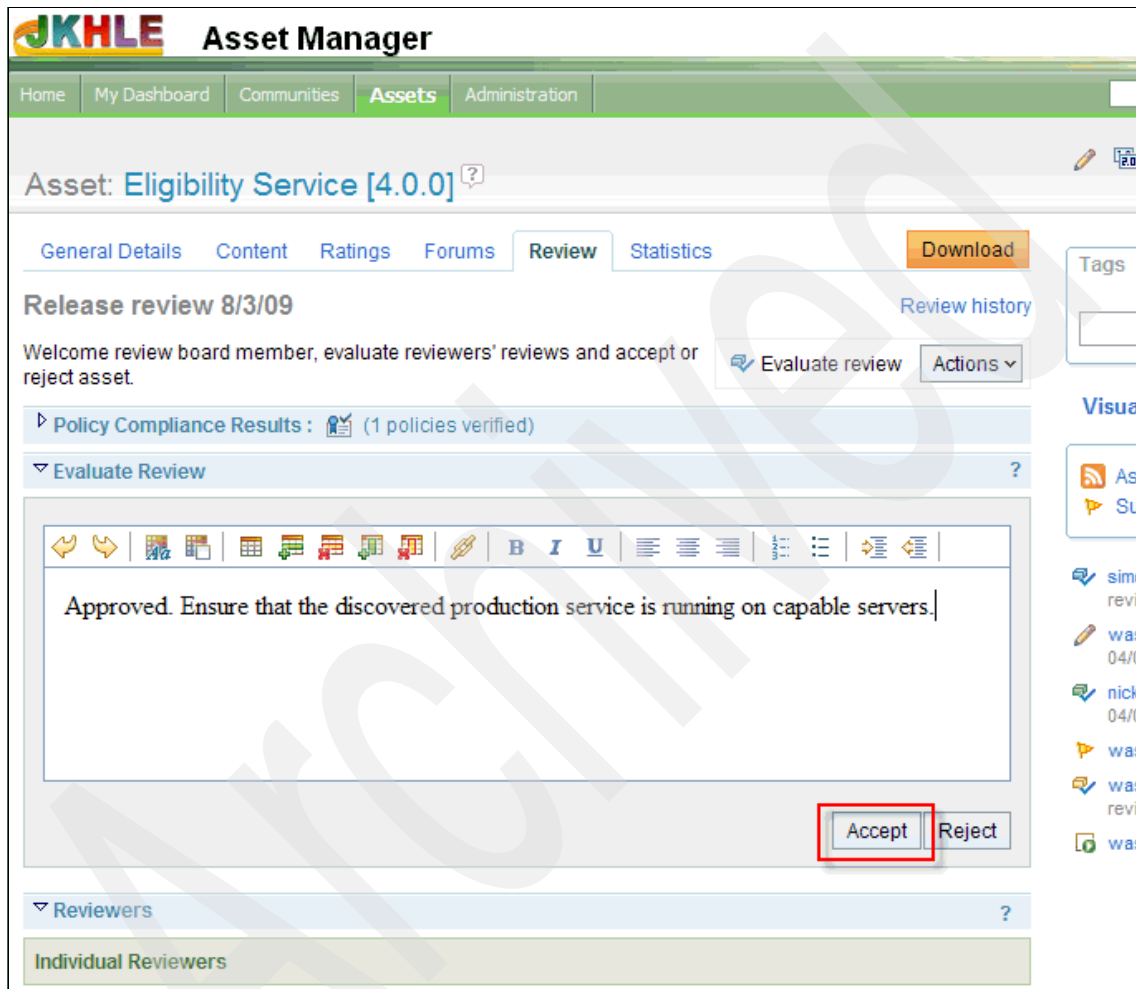


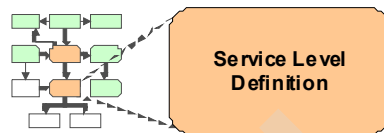
Figure 12-5 Shared Services release manager has reviewed the Release Asset and accepts it

After the acceptance and Advanced Lifecycle Edition synchronization, the entity is placed in the Planned state within WSRR.

## 12.5 Enabling governed consumption and reuse

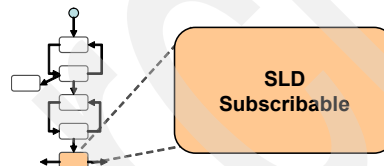
Now that the scoping and planning information is complete, we need to detail the consumption specification of the Eligibility service that we have exposed in WSRR Advanced Lifecycle Edition. Providing a specification and Service Level Definition (SLD) for the Eligibility service enables governed consumption and reuse.

The specification details include a clear statement about the service levels that consumers of this service can access. The SLD asset is the entity that is used to contain this information. It provides access to key information classes that are extracted from the WSDL when it is loaded.



The development team initiates the process of creating the SLD and populates its specification within WSRR. A default SLD can be created for the Eligibility service by using the New SLD button on its Details view.

The steps described in 8.2, “Creating a Service Level Definition” on page 186 identify what information is required on the SLD and move it through to the SLD Subscribable state. The only exception to those steps is to add the Eligibility service’s service endpoint when the SLD is in the SLD Scoped state. The Eligibility service’s service endpoint can be added, because the WSDL for the production Eligibility service has already been loaded.



*Service Level Definition Lifecycle – Model & Assemble*

The new SLD looks similar to the SLD that is shown in Figure 12-6 on page 285, because it has been made subscribable.



JKHLE

Service Registry and Repository

Perspective: Development ▾

Home ▾

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Extended Service Level Definition

Service Level Definitions > SLD - Eligibility Service

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

Name

SLD - Eligibility Service

Description

Defines the QoS for the production Eligibility service that was discovered running on server X on 20 July 09.

Average Response Time

10ms

Availability

24/7 High Availability

Additional Properties

Back

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

Eligibility\_Service

Available Endpoints

<http://www.jkhle.com/jkhle/services/Eligibility>

Bound SCA Export

None

Bound Web Service Port

EligibilityPort

Compatible Service Level Definitions

None

Dependent Entities

Consuming Service Level Agreement(s)

None

Providing Capability Version

Eligibility Service

Governance State

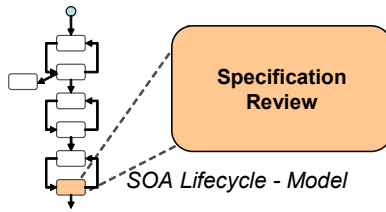
SLD Subscribable

Figure 12-6 Service Level Definition is now subscribable

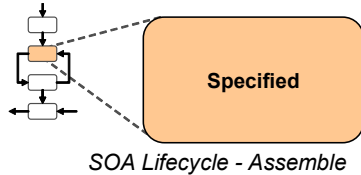
Development Lead

The SLD is ready and set up for consumers to subscribe to it. The owning Eligibility service version has been fully specified with the relevant quality of service (QoS) information, which is contained within the SLD.

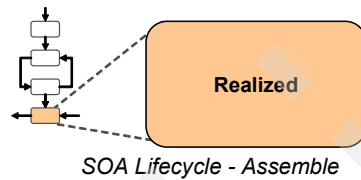
Chapter 12. Bottom-up scenario 285



The CoE team needs to review the owning Eligibility service version in order for the development team to move the service version for the Specification Review.



The CoE team reviews the new Eligibility service version and transitions it to the Specified state.



To complete the realization of the service version, any deployed or installed assets can now be associated to the service version prior to the service version being put up for review within the development team. The development lead and the development release managers agree to the details and move the service version into the Realized state.

## 12.6 Completing the SOA service lifecycle for the Eligibility service

The Eligibility service version must be moved from the Realized state through the relevant deployment stages in the SOA service lifecycle so that it is registered as Operational. These lifecycle stages are present so that while the service is developed in a top-down manner, it can be promoted to the various staging environments and tested appropriately.



The Shared Services operations team promotes the service version and its related configurations into the staging (as known as preproduction) and production registries. To achieve the move into the staging environment, the service endpoint must be tagged as being in the Staging environment, declared as being Online, proposed for Staging, and then, approved for Staging.

There are four stages involved here:

1. Classify the Eligibility service endpoint as Staging. Figure 12-7 shows this activity.

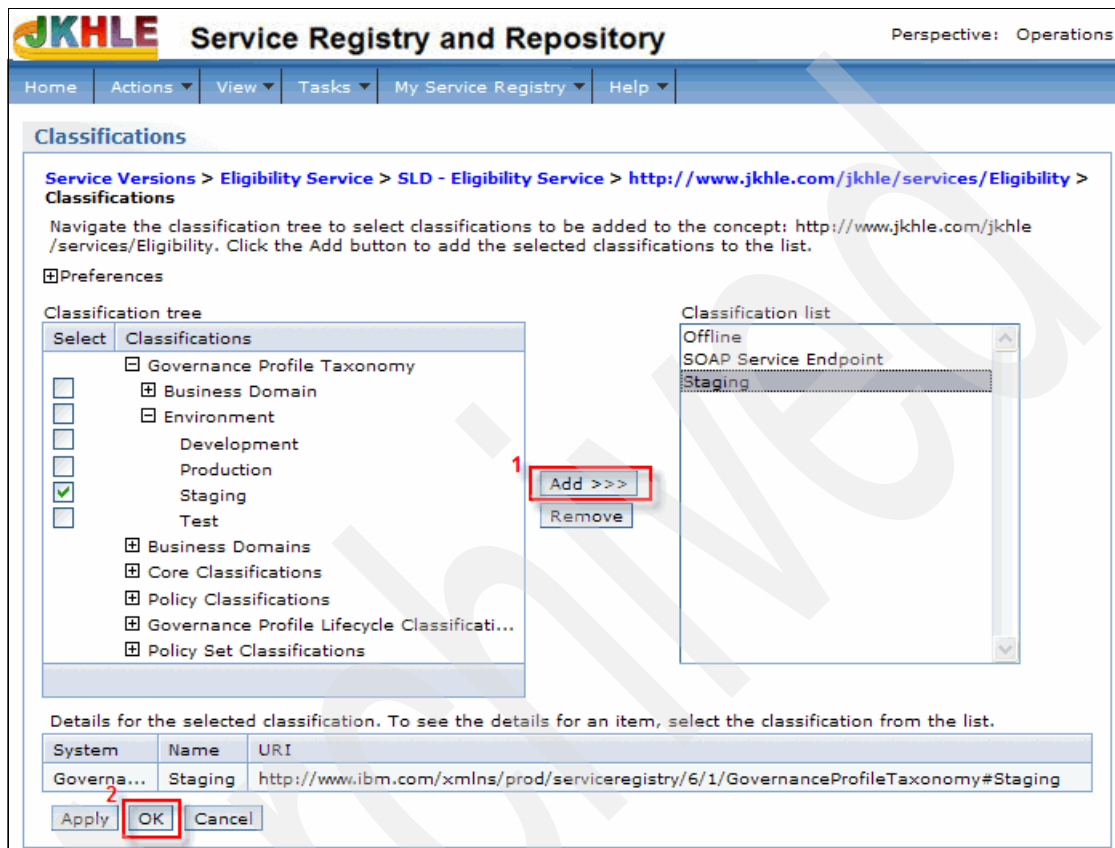


Figure 12-7 Declare the service endpoint as ready for Staging

2. Transition the service endpoint into the Online state. Figure 12-8 on page 288 shows this activity.

JKHLE

Service Registry and Repository

Perspective: Operations

[Home](#)
[Actions](#)
[View](#)
[Tasks](#)
[My Service Registry](#)
[Help](#)

SOAP Service Endpoint

[Service Versions](#) >
[Eligibility Service](#) >
[SLD - Eligibility Service](#) >
<http://www.jkhle.com/jkhle/services/Eligibility>

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

Details
 Impact Analysis
 Governance
 Policy
 Activity

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

Properties

\*Name

http://www.jkhle.com/jkhle/services/Eligibility

Description

Namespace

http://jkhle.com/Eligibility\_Service

Owner

debra

Version

4.0.0

Last modified

Tuesday, 4 August 2009 14:23:08 ET

\*Endpoint Type

SOAPAddress

\*Port Name

EligibilityPort

Service Version

4.0.0

\*Service Namespace

http://jkhle.com/Eligibility\_Service

\*Service Name

EligibilityService

Additional Properties

Back

Approve For Use

Retire From Use

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Soap Address

EligibilityPort\_SOAPAddress

WSDL Ports

EligibilityPort

Source Document

EligibilityService.wsdl

Governance State

Offline

Classifications

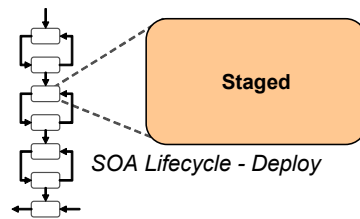
Offline

SOAP Service Endpoint

Staging

Figure 12-8 Approve the service endpoint for use, which places it into the Online state

- The operations team moves back to the Eligibility service version and propose staging deployment. The team can now check that the information in the governance registry is correct for deployment to the staging registry.

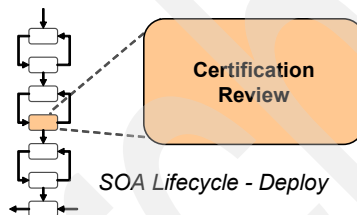


4. After checking the information, the operations team can approve the service version for staging.

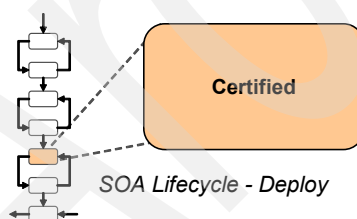
**Note:** When promotion in WSRR is enabled, the promoted entity and all of its dependent entities will be promoted into the staging registry (provided that an environment classification is not set or the environment classification matches the one defined in the promotion configuration). This functionality is known as *promotion filtering*, which is described in the WSRR Information Center at this Web site:

[http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/cwsr\\_promotion\\_filtering.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp?topic=/com.ibm.sr.doc/cwsr_promotion_filtering.html)

The operations team certifies that the service endpoint performs correctly and tests that the Eligibility service will meet its specified SLDs if and when the Eligibility service is promoted to the production environment.



The operations team selects the Propose Certification button for Certification Review.



The CoE team can perform the approval of the proposed certification to place it in the Certified state.

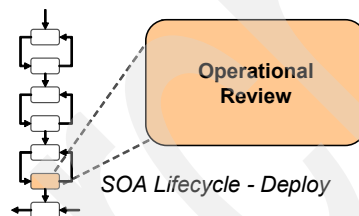
**Note:** Even though the service endpoint has been promoted to the staging registry, there will not be any supporting service implementation available because it was discovered in the production environment.

Therefore, before the service endpoint can be certified, the operations and development teams need to make the discovered Eligibility service implementation available within the staging environment. That way, other teams can develop and test against the Eligibility service in staging and not have to rely on the implementation in production.

However, if you do not want the implementation duplicated in the staging environment, you must perform these stages in order to get through the staging and certification lifecycle states in WSRR. Or, you must customize the lifecycle to transition past them.



Finally, it is not necessary to deploy an implementation to production, because the service implementation is already shared. However, the endpoint and supporting service version and SLD information must be made available within the production registry. The service endpoint environment classification needs to be updated, by the operations team, for Production use. To do so, the operations team edits the classifications on the Eligibility service endpoint, removes the Staging classification, and adds the Production classification.



The operations team now moves back to the Eligibility service version and select Propose Production Deployment for operational review as shown in Figure 12-9 on page 291.

JKHLE

Service Registry and Repository

Perspective: Operations ▾

Home ▾

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Service Version

Service Versions > Eligibility Service

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Eligibility Service

Description

This service identifies whether a customer satisfies the commercial credentials to be an account holder.

Version

4.0.0

Consumer Identifier

SSES040

Version Availability Date

Monday, 1 June 2009

Version Termination Date

Monday, 31 December 2012

Version Requirements Link

<http://www.jkhle.com/db/requirements/eligibility.doc>

Asset Web Link

<https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B0606A66C-74F4-202F-B3B6-635AAE288DDA%7D&v=4.0.0>

Remote State

Approved

Owner Email

connie@jkhle.com

Additional Properties

Back

New Version

Propose Production Deployment

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

EligibilityService

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

EligibilityService.wsdl

Provides

SLD - Eligibility Service

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Account eligibility

Consumer DOU(s)

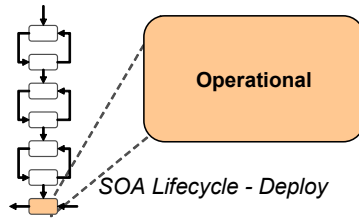
None

Provider DOU(s)

None

Governance State

Figure 12-9 Operations team propose the service version for production use



It is now up to the Shared Services business unit leader to approve the service for production use by viewing the service version and clicking Approve Production Deployment to make it operational. This action initiates the promotion of the service version and all its dependent production entities into the production registry.

Our bottom-up process is now complete. We have taken the discovered Eligibility service definition of the implementation running in a production environment, described it in WSRR Advanced Lifecycle Edition, and registered it in the governance and production registries so that other teams can now discover and subscribe to it.



# Endpoint management

This chapter shows the WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition enabling JKHLE Enterprises (JKHLE) to enforce service governance around endpoint management.

This chapter includes these topics:

- ▶ 13.1, “Introduction” on page 294
- ▶ 13.2, “Problem analysis and defect resolution” on page 294
- ▶ 13.3, “Staging deployment” on page 295
- ▶ 13.4, “Production deployment” on page 299

## 13.1 Introduction

In this scenario, the Sporting Goods Ordering application development team has identified a defect in the Execute Payment service that they consume. In order to address this problem, the Sporting Goods development team has raised a defect tracking ticket with the Shared Services development team that owns the issue.

The following sections of this chapter describe the steps involved in developing a fix for this issue and then moving this fix through the service governance lifecycle so that it is fully deployed to production.

## 13.2 Problem analysis and defect resolution

Problems must be analyzed, and a resolution must be identified and tested before a fix can be put in production.



### Collecting diagnostic data

The Shared Services development team first tries to reproduce the problem within the test environment. The team identifies the environmental triggers and successfully creates a scenario in which to reliably reproduce the problem. The test team uses the tracing facilities of the service implementation to gather as much information about the causes as possible.



### Examining the diagnostics and determining a solution

The assigned Shared Services developer examines the trace and confirms that the cause of the problem lies within the Execute Payment service version that is deployed to production. The developer designs a solution, which is discussed with colleagues to verify that there will not be any unforeseen side effects.



### Testing the solution

When the developer concludes that the proposed fix is suitable, the developer implements the change and engages the test team to verify the changes and to run the test team's suite of regression tests. When these tests are complete, the test lead identifies an implementation release candidate.



### Approving the solution

The release manager of the Shared Services organization reviews the proposed realization. After reviewing the status with the development and test teams and after confirming that the required development and test processes have been

completed, the release manager approves the proposed realization and notifies the operations manager of the availability.

## 13.3 Staging deployment



The Shared Services operations team lead deploys the proposed build candidate to the staging environment, following the JKHLE standard deployment procedures to identify the assets to deploy, and undertakes the deployment process. After deployed the service and successfully completing the deployment verification tests, the operations team lead retrieves the Web Services Description Language (WSDL) for the deployed service and loads this WSDL into WSRR, as shown in Figure 13-1.



Figure 13-1 Shared Services operations lead loads the WSDL containing the staging endpoints for the Execute Payment service version

After loading the new WSDL into WSRR, the operations lead classifies the new staging endpoint object appropriately for the environment. After loading and classifying the service version endpoint, the operations lead needs to build the relationship between the Execute Payment service and the service version endpoint that was just loaded and classified, as shown in Figure 13-2 on page 296.

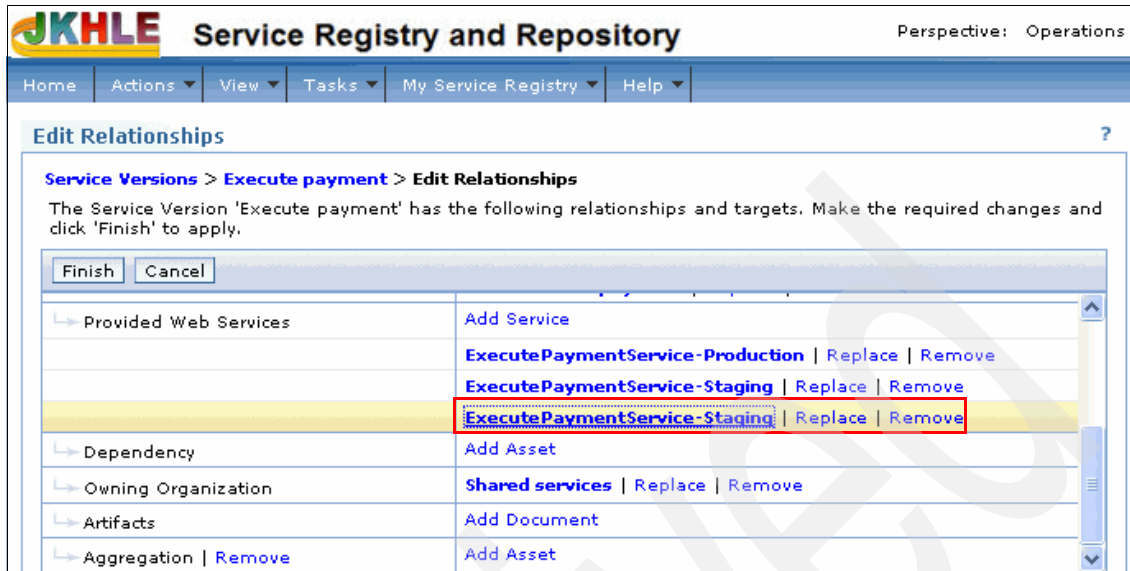


Figure 13-2 Shared Services operations lead relates the staging endpoints with the Execute Payment service version

After loading and classifying the endpoint in WSRR, the operations lead builds the relationship between the Service Level Definition (SLD) and the staging endpoint, as shown in Figure 13-3.

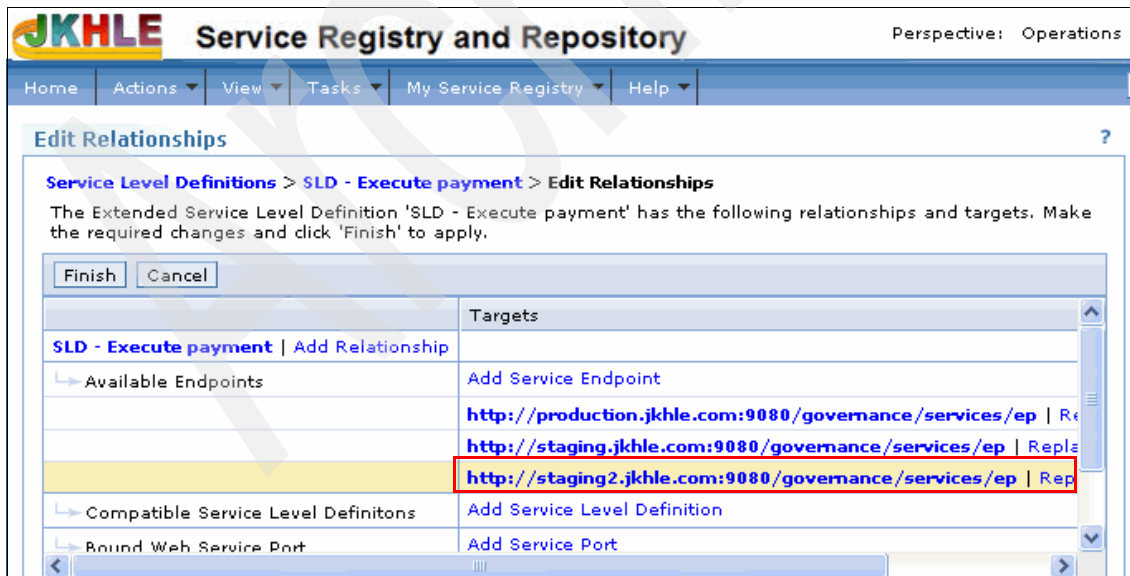
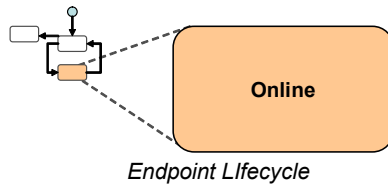


Figure 13-3 Shared Services operations lead relates the staging endpoints to the Service Level Definition



The operations lead performs a final verification that the service has been deployed successfully to the staging environment and that it is online. This verification is a check that the service is running, not that it functions correctly. After verifying that the service is available, the operations lead approves the service endpoint for use to indicate that it is online to the service consumers, as shown in Figure 13-4 on page 298.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

SOAP Service Endpoint

Service Level Definitions > SLD - Execute payment > http://staging2.jkhle.com:9080/governance/services/ep

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

http://staging2.jkhle.com:9080/governance/services/ep

Description

Namespace

http://www.jkhle.com/governance/service/ep

Owner

jon

Version

1.0.4

Last modified

Tuesday, 4 August 2009 18:04:45 ET

\*Endpoint Type

SOAPAddress

\*Port Name

ExecutePaymentPort

Service Version

1.0.4

\*Service Namespace

http://www.jkhle.com/governance/service/ep

\*Service Name

ExecutePaymentService-Staging

Additional Properties

Back

Approve For Use

Retire From Use

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Soap Address

ExecutePaymentPort\_SOAPAddress

WSDL Ports

ExecutePaymentPort

Source Document

ExecutePaymentStaging2.wsdl

Governance State

Offline

Classifications

Offline

SOAP Service Endpoint

Staging

Figure 13-4 Shared Services operations lead approves the staging endpoints for use

### 13.3.1 Staging review

The Shared Services operations team and test team execute a complete set of regression tests, as well as service version-specific test scenarios. These tests

verify that the service performs to both the functional and nonfunctional details of the specification, particularly ensuring that the deployed instance meets the requirements of the Service Level Definition. After completing their final verification testing of the service, the Shared Services teams approve the staging deployment.

## 13.4 Production deployment

The Execute Payment service version is now ready to be deployed to production. The Shared Services operations team lead deploys the proposed build candidate to the production environment. After deploying the service version and successfully completing the deployment verification tests, the operations lead retrieves the WSDL for the deployed service version and loads this WSDL into WSRR, as shown in Figure 13-5.



Figure 13-5 Shared Services operations lead loads the WSDL containing the production endpoints

After loading the new WSDL into WSRR, the operations lead classifies the new production endpoint object appropriately for the environment.

After loading and classifying the service version endpoint, the operations lead needs to build the relationship between the Execute Payment service version and the service version endpoint that was just loaded and classified, as shown in Figure 13-6 on page 300.

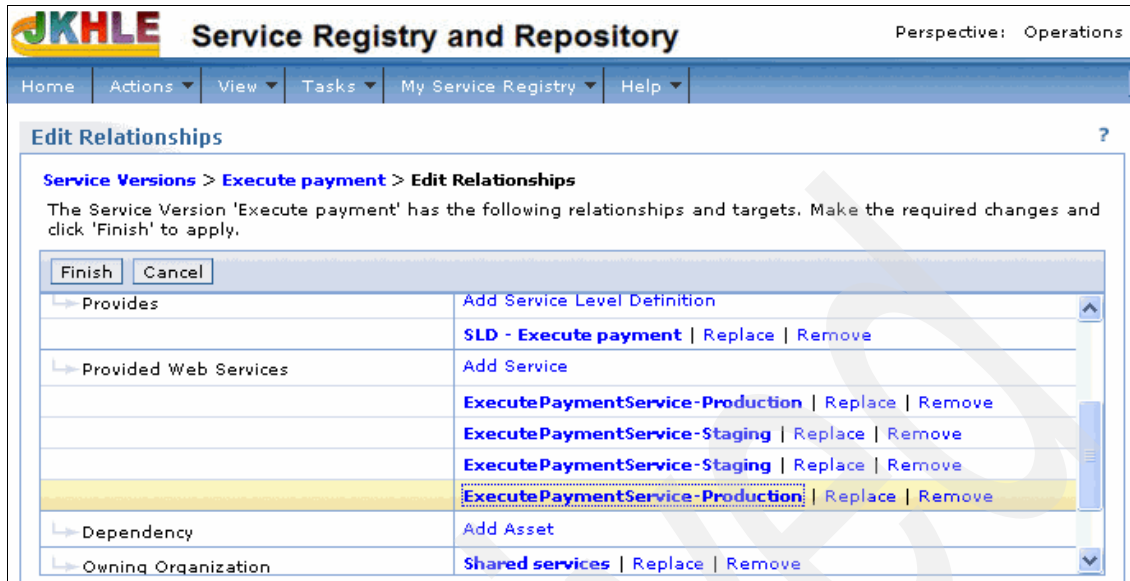


Figure 13-6 Shared Services operations lead relates the Execute Payment service version with the production endpoints

After loading and classifying the endpoint in WSRR, the operations lead builds the relationship between the Service Level Definition and the production endpoint, as shown in Figure 13-7 on page 301.



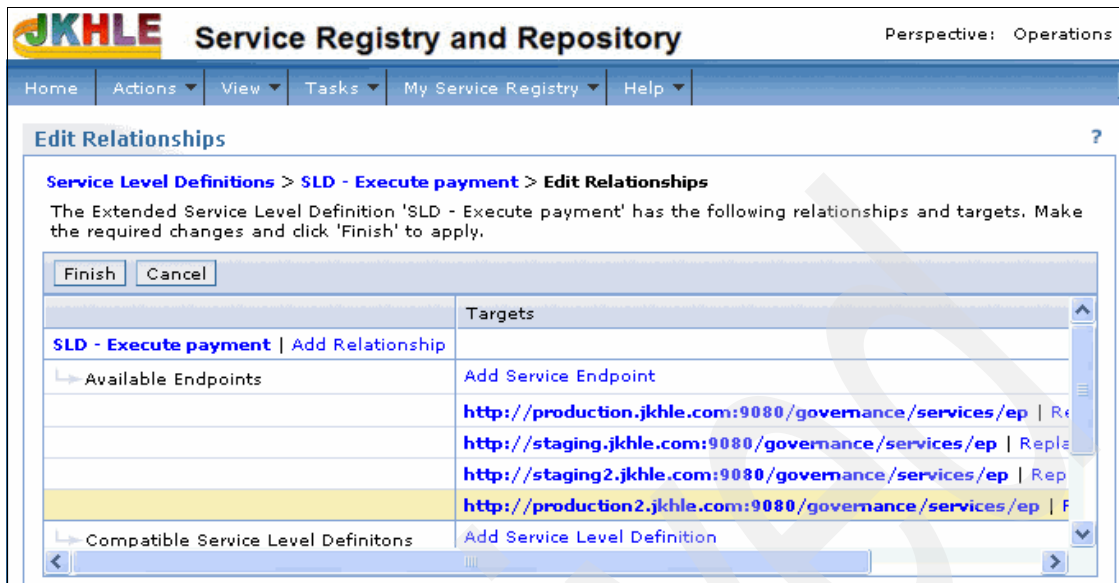
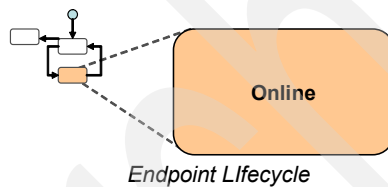


Figure 13-7 Shared Services operations lead relates the Service Level Definition with the production endpoints



The operations lead performs a final verification that the service has been deployed successfully in the production environment and that it is online. After verifying that the service is available, the operations lead approves the service endpoint for use to indicate that it is online to the service consumers, as shown in Figure 13-8 on page 302.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

SOAP Service Endpoint

Service Level Definitions > SLD - Execute payment > http://production2.jkhle.com:9080/governance/services/ep

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

http://production2.jkhle.com:9080/governance/services/ep

Description

Namespace

http://www.jkhle.com/governance/service/ep

Owner

jon

Version

1.0.4

Last modified

Tuesday, 4 August 2009 18:06:58 ET

\*Endpoint Type

SOAPAddress

\*Port Name

ExecutePaymentPort

Service Version

1.0.4

\*Service Namespace

http://www.jkhle.com/governance/service/ep

\*Service Name

ExecutePaymentService-Production

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Soap Address

ExecutePaymentPort\_SOAPAddress

WSDL Ports

ExecutePaymentPort

Source Document

ExecutePaymentProduction2.wsdl

Governance State

Offline

Classifications

Offline

Production

SOAP Service Endpoint

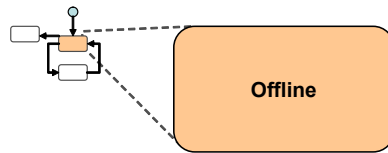
Additional Properties

Back

Approve For Use

Retire From Use

Figure 13-8 Shared Services operations lead approves the production endpoint for use



Endpoint Lifecycle

The final step in the process is to remove the old production service from use. To do so, the operations lead revokes the old production endpoint from use, as shown in Figure 13-9 on page 303.

**Service Registry and Repository**
Perspective: Operations | jon

[Home](#)
[Actions](#)
[View](#)
[Tasks](#)
[My Service Registry](#)
[Help](#)

### SOAP Service Endpoint

**Service Level Definitions** > **SLD - Execute payment** > **http://production.jkhle.com:9080/governance/services/ep**

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

[Details](#)
[Impact Analysis](#)
[Governance](#)
[Policy](#)
[Activity](#)

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

**Properties**

- \*Name  
http://production.jkhle.com:9080/governance/services/ep
- Description
- Namespace  
http://www.jkhle.com/governance/service/ep
- Owner  
jon
- Version  
1.0
- Last modified  
Tuesday, 4 August 2009 18:10:19 ET
- \*Endpoint Type  
SOAPAddress
- \*Port Name  
ExecutePaymentPort
- Service Version  
1.0
- \*Service Namespace  
http://www.jkhle.com/governance/service/ep
- \*Service Name  
ExecutePaymentService-Production

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

- Soap Address  
ExecutePaymentPort\_SOAPAddress
- WSDL Ports  
ExecutePaymentPort
- Source Document  
ExecutePaymentProduction.wsdl

**Governance State**

- Online

**Classifications**

- Production
- SOAP Service Endpoint
- Online

**Additional Properties**

[Back](#)
[Revoke From Use](#)

**Help**  
 For field information, click on the field label when the tooltip appears.

Figure 13-9 Operations lead revokes the old production endpoint from use

After replacing the production endpoint, the Sporting Goods Ordering application will not encounter the issue that triggered this process. The Shared Services team returns the defect tracking ticket to the originator of the problem. The originator closes the problem when the originator is satisfied that the situation is resolved.

# Service Level Definition management

This chapter describes how WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition enables JKHLE Enterprises (JKHLE) to enforce service governance around the management of subscribable Service Level Definitions.

This chapter discusses these topics:

- ▶ 14.1, “Introduction” on page 306
- ▶ 14.2, “Scenario” on page 306
- ▶ 14.3, “Consumer identification” on page 306
- ▶ 14.4, “Superseding the service version” on page 307
- ▶ 14.5, “Consumer response to notification” on page 310
- ▶ 14.6, “Deprecating the Service Level Definition” on page 321
- ▶ 14.7, “Retiring a Service Level Definition” on page 324

## 14.1 Introduction



After a service implementation has been deployed to a staging or production environment, the service implementation likely has service consumers. The management of the relationships with these consumers and the migration of these consumers onto alternative providers are important components of service governance.

This chapter describes how (WSRR) Advanced Lifecycle Edition supports the management of these situations.

## 14.2 Scenario

In this scenario, the JKHLE Shared Services team has two versions of the Ship Goods service that are currently available for consumers to subscribe to and invoke. The Shared Services release manager determines that, because that the latest version of the service is backward compatible, migrating existing consumers of the older version of the service to the newer version will be a significant cost savings.

## 14.3 Consumer identification

The first step in managing the deprecation of a production service version is for the development team to understand the impact of the proposed change. The Shared Services release manager uses the impact analysis capabilities of WSRR to understand the potential impact on service consumers. Figure 14-1 shows the impact analysis graph view of the consumer services that are dependent on the Ship Goods service version that is being assessed.

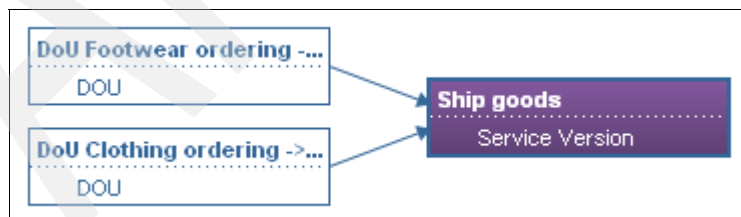


Figure 14-1 The two DOUs that depend on the Ship Goods service version

From this graphical view, the release manager can see that there are two business capabilities that will be impacted by this proposed change: Footwear ordering and Clothing ordering.

**Tip:** If the release manager identifies that there are no service level agreements (SLAs) that directly reference this Service Level Definition (SLD), they can immediately deprecate the service version, as described in 14.6, “Deprecating the Service Level Definition” on page 321.


### 14.3.1 Consumer notification

The owning organizations of the consuming applications need to be notified of the proposed change. When JKHL Enterprises configured WSRR, the company created a set of triggers and actions that automatically creates subscriptions for the owners of subscribed applications, as detailed in 4.2.6, “Customizations” on page 89. For this reason, JKHLE needs only to transition the service version to the appropriate state in the lifecycle, and all of the service consumer owners will be automatically notified.

## 14.4 Superseding the service version



Before indicating that the service version is superseded, the operations lead reviews all of the Ship Goods service version’s Service Level Definitions to ensure that all of them reference all compatible SLDs. Figure 14-2 on page 308 shows the operations lead reviewing the relationships in the SLD of the old Ship Goods service version.



Service Registry and Repository

Perspective: Operations

Home
Actions
View
Tasks
My Service Registry
Help

Extended Service Level Definition

Service Level Definitions > SLD - Ship goods

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details
Impact Analysis
Governance
Policy
Activity

Edit Properties
Edit Relationships
Edit Classifications

Properties

\*Name

SLD - Ship goods

Description

Specifies the level of service which will be provided as standard to the Web site

Average Response Time

4 seconds

Availability

24/7 High Availability

Additional Properties

Back

Supercede

Deprecate

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

ShipGoods

Available Endpoints

<http://production.jkhle.com:9080/governance/services/sg>
<http://staging.jkhle.com:9080/governance/services/sg>

Bound SCA Export

None

Bound Web Service Port

ShipGoodsPort

Compatible Service Level Definitions

SLD - Ship goods version 2.1

Dependent Entities

Consuming Service Level Agreement(s)

[SLA - DoU Clothing ordering -> Ship goods](#)
[SLA - DoU Footwear ordering -> Ship goods](#)

Providing Capability Version

Ship goods

Governance State

SLD Subscribable

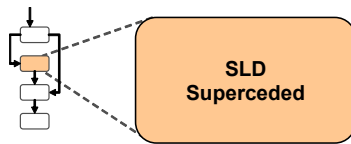
Classifications

SLD Subscribable

Extended Service Level Definition

Figure 14-2 Operations lead reviews the Service Level Definition





Service Level Definition Lifecycle –  
Deploy & Manage

The operations lead transitions the old version of the Ship Goods service to the superseded state, as shown in Figure 14-3.

**JKHLE Service Registry and Repository** Perspective: Operations

Home Actions View Tasks My Service Registry Help

**Extended Service Level Definition**

**Service Level Definitions > SLD - Ship goods**

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details Impact Analysis Governance Policy Activity

Edit Properties | Edit Relationships | Edit Classifications

**Properties**

\*Name  
SLD - Ship goods

Description  
Specifies the level of service which will be provided as standard to the Web site

Average Response Time  
4 seconds

Availability  
24/7 High Availability

**Additional Properties**

Back **Supersede** Deprecate

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

Service Interface  
ShipGoods

Available Endpoints

- <http://production.jkhle.com:9080/governance/services/sg>
- <http://staging.jkhle.com:9080/governance/services/sg>

Bound SCA Export  
None

Bound Web Service Port  
ShipGoodsPort

Figure 14-3 Operations lead supersedes the Service Level Definition

The action of superseding the service version automatically generates an e-mail notification to all of the service consumers.

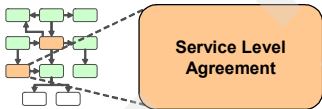
In this state, the owner of the service version indicates to the service consumers that they need to negotiate new service level agreements with the alternative Service Level Definition providers. The operations lead has assisted the consumers in this process by ensuring that the Service Level Definition that is being superseded has a relationship to any alternative Service Level Definitions.

Potential new Service Level Definition consumers will not be able to create new service level agreement relationships to a Service Level Definition that is in this superseded state.

## 14.5 Consumer response to notification

When a consuming application owner is notified of the superseding of a Service Level Definition, the consuming application owner needs to take action to avoid an interruption in the quality of service that the consuming application receives. The recipient views the details of the Service Level Definition that is being superseded and examines the capabilities of the alternative Service Level Definitions and the service versions that underpin these capabilities.

### 14.5.1 Consumer creates new service level agreement

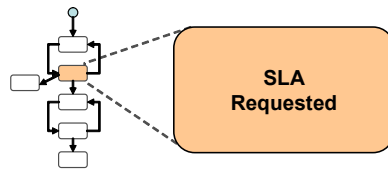


The development lead for the Footwear Ordering application identifies the latest version of the Ship Goods SLD as the suitable replacement for the SLD that is being superseded. The development lead creates a new service level

agreement to the latest Ship Goods Service Level Definition, as shown in Figure 14-4.



Figure 14-4 Footwear Ordering application development lead creates a new Service Level Agreement



The Footwear Ordering application development lead proposes the service level agreement for review and approval by the provider organization, as shown in Figure 14-5 on page 312.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Agreement

DOUs > DoU Footwear ordering -> Ship goods > SLA - DoU Footwear ordering -> Ship goods

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLA - DoU Footwear ordering -> Ship goods

Description

Context Identifier

Service Level Agreement Availability Date

Tuesday, 4 August 2009

Service Level Agreement Termination Date

Saturday, 1 September 2012

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

20

Peak Message Rate Daily Time

17:00

Peak Message Rate Daily Duration

60

Additional Properties

Back

Request SLA

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Ship goods version 2.1

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Footwear ordering

Governance State

SLA Identified

Classifications

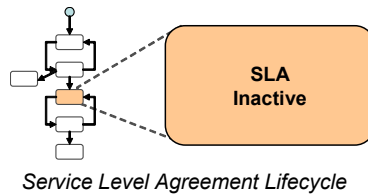
SLA Identified

Extended Service Level Agreement

Figure 14-5 Footwear Ordering application development lead requesting the new Service Level Agreement

312 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition

## 14.5.2 Service Level Agreement approval



The Footwear Ordering application development lead notifies the provider that there is a proposed SLA for the provider's review. The Shared Services development lead for the Ship Goods service reviews and approves the specified SLA, as shown in Figure 14-6 on page 314.

JKHLE

Service Registry and Repository

Perspective: Development

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Agreement

DOUs > DoU Footwear ordering -> Ship goods > SLA - DoU Footwear ordering -> Ship goods

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLA - DoU Footwear ordering -> Ship goods

Description

Context Identifier

Service Level Agreement Availability Date

Tuesday, 4 August 2009

Service Level Agreement Termination Date

Saturday, 1 September 2012

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

20

Peak Message Rate Daily Time

17:00

Peak Message Rate Daily Duration

60

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Ship goods version 2.1

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Footwear ordering

Governance State

SLA Requested

Classifications

SLA Requested

Extended Service Level Agreement

Additional Properties

Back

Revise SLA Request

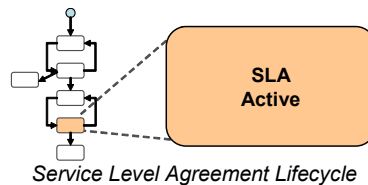
Reject SLA Request

Approve SLA Request

Figure 14-6 Shared Services development lead approves the new Service Level Agreement

After its approval, this SLA moves into the Inactive state. The Shared Services team has agreed in principle to provide access to the specific Service Level Definition (and therefore, to provide the quality of service that it defines), but the Shared Services team has not yet provided the authorization to access any of the endpoints. The team might delay the authorization for a number of reasons, for example, the Shared Services team might need to deploy new instances of the endpoint to provide the required level of service.

### 14.5.3 Service Level Agreement activation



The Shared Services operations lead reviews the capacity that is deployed within the staging and production environments. When the Shared Services operations lead has confirmed that the required capacity is in place, the Shared Services operations lead classifies the SLA with the environment in

which the consumer is to be allowed access and activates the SLA. Figure 14-7 on page 316 shows the Shared Services operations lead approving the SLA for use, after classifying it as available in the Staging environment.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Agreement

Service Level Agreements > SLA - DoU Footwear ordering -> Ship goods

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLA - DoU Footwear ordering -> Ship goods

Description

Context Identifier

Service Level Agreement Availability Date

Tuesday, 4 August 2009

Service Level Agreement Termination Date

Saturday, 1 September 2012

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

20

Peak Message Rate Daily Time

17:00

Peak Message Rate Daily Duration

60

Additional Properties

Back

Activate SLA

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Ship goods version 2.1

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Footwear ordering

Governance State

SLA Inactive

Classifications

SLA Inactive

Extended Service Level Agreement

Figure 14-7 Shared Services operations lead activates the new Service Level Agreement

This activation allows the consumers to begin their testing process by enabling the JKHLE enterprise service bus (ESB) to select the new service level

316 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition



agreement based on the environment classification, thereby invoking the new version of the service in the staging environment.

#### 14.5.4 Integration testing



The test team for the Footwear Ordering application undertakes the standard JKHLE standard set of integration and regression testing against the new Ship Goods service version in the staging environment.



After completing the testing process, the operations lead for the Footwear Ordering application confirms the operations lead's acceptance of the new service level agreement with the Shared Services operations team lead.

#### 14.5.5 Production release



The Shared Services operations team lead classifies the service level agreement as being available in the Production environment, as shown in Figure 14-8 on page 318.

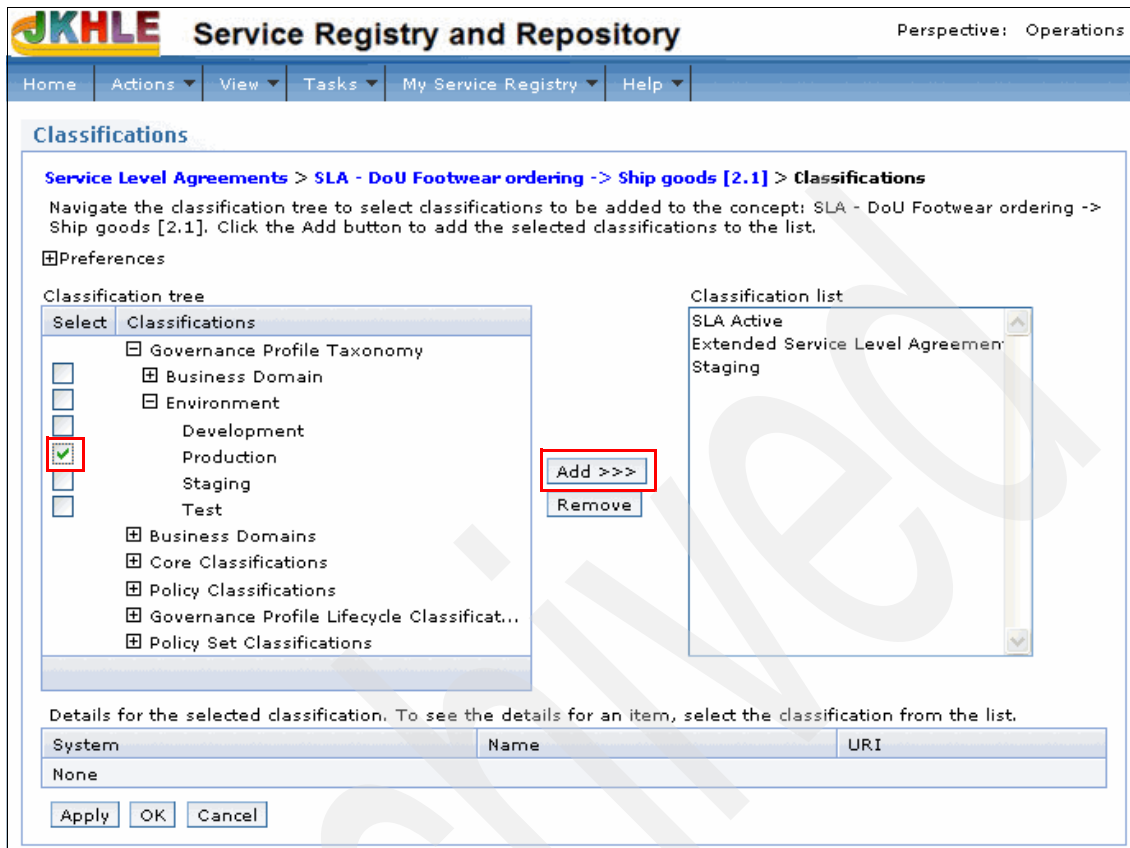
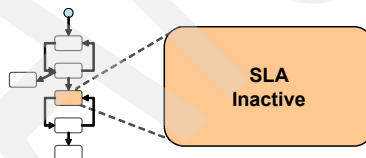


Figure 14-8 Operations lead classifies the Service Level Agreement in the Production environment

This classification allows the consumers to invoke the new version of the service in the production environment.



Service Level Agreement Lifecycle

The Footwear operations lead deactivates the old SLA in order to ensure that all new application traffic utilizes the new Ship Goods service version, as shown in Figure 14-9 on page 319.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Extended Service Level Agreement

Service Level Agreements > SLA - DoU Footwear ordering -> Ship goods

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

SLA - DoU Footwear ordering -> Ship goods

Description

Context Identifier

FOSV010

Service Level Agreement Availability Date

Tuesday, 4 August 2009

Service Level Agreement Termination Date

Tuesday, 4 August 2009

Version Match Criteria

LatestCompatibleVersion

Average Messages Per Day

200

Maximum Messages Per Day

500

Minimum Messages Per Day

50

Peak Message Rate

20

Peak Message Rate Daily Time

17:00

Peak Message Rate Daily Duration

60

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Agreed Endpoints (Service Level Definitions)

SLD - Ship goods

Service Level Policies

None

Bound SCA Import

None

Dependent Entities

Consuming Capability Version

Footwear ordering

Governance State

SLA Active

Classifications

SLA Active

Extended Service Level Agreement

Additional Properties

Back

Deactivate SLA

Figure 14-9 Footwear operations lead deactivates the Service Level Agreement

After a period of time of successful consumption of the new Ship Goods service version, which is determined by JKHL Enterprises' governance processes, the

SOA Governance CoE representative can terminate the SLA, as shown in Figure 14-10, in order to remove the SLA completely from service.

**JKHLE Service Registry and Repository** Perspective: SOA Governance ▾

Home Actions ▾ View ▾ Tasks ▾ My Service Registry ▾ Help ▾

### Extended Service Level Agreement

**Service Level Agreements > SLA - DoU Footwear ordering -> Ship goods**

Detail view for Service Level Agreement. The Service Level Agreement (SLA) represents the details of a specific service subscription and defines the quality of service properties that will be used in any interactions between the consuming capability version and the agreed provider endpoints.

Details Impact Analysis Governance Policy Activity

[Edit Properties](#) | [Edit Relationships](#) | [Edit Classifications](#)

#### Properties

**\*Name**  
SLA - DoU Footwear ordering -> Ship goods

**Description**

**Context Identifier**

**Service Level Agreement Availability Date**  
Tuesday, 4 August 2009

**Service Level Agreement Termination Date**  
Saturday, 1 September 2012

**Version Match Criteria**  
LatestCompatibleVersion

**Average Messages Per Day**  
200

**Maximum Messages Per Day**  
500

**Minimum Messages Per Day**  
50

**Peak Message Rate**  
20

**Peak Message Rate Daily Time**  
17:00

**Peak Message Rate Daily Duration**  
60

#### Additional Properties

[Back](#) [Terminate SLA](#)

#### Links

- [Graphical View](#)
- [Applied Policies](#)
- [Applied Policy Attachments](#)

#### Relationships

Agreed Endpoints (Service Level Definitions)  
[SLD - Ship goods version 2.1](#)

Service Level Policies  
None

Bound SCA Import  
None

#### Dependent Entities

Consuming Capability Version  
[Footwear ordering](#)

#### Governance State

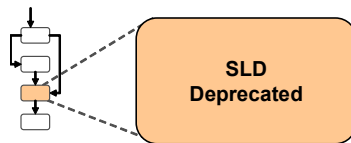
SLA Inactive

#### Classifications

SLA Inactive  
Extended Service Level Agreement

Figure 14-10 CoE representative terminates the SLA

## 14.6 Deprecating the Service Level Definition



*Service Level Definition Lifecycle –  
Deploy & Manage*

After all of the consumers of the old Ship Goods service version's SLD have been migrated to the new SLD, the Shared Services operations lead can deprecate the SLD. The Shared Services operations lead can verify that all of the consumers have been migrated by verifying the state of the service level agreements that consume the SLD in question.

The Shared Services operations lead determines that all of the consumers have been migrated and so deprecates the Ship Goods service version, as shown in Figure 14-11.

**JKHLE Service Registry and Repository** Perspective: Operations

Home Actions View Tasks My Service Registry Help

### Extended Service Level Definition

**Service Level Definitions > SLD - Ship goods**

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details Impact Analysis Governance Policy Activity

Edit Properties | Edit Relationships | Edit Classifications

**Properties**

- Name: SLD - Ship goods
- Description: Specifies the level of service which will be provided as standard to the Web site
- Average Response Time: 4 seconds
- Availability: 24/7 High Availability

**Additional Properties**

Back Deprecate

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

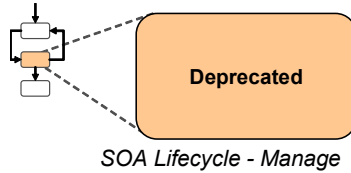
- Service Interface: ShipGoods
- Available Endpoints:
  - <http://production.jkhle.com:9080/governance/services/sg>
  - <http://staging.jkhle.com:9080/governance/services/sg>
- Bound SCA Export: None
- Bound Web Service Port: ShipGoodsPort

*Figure 14-11 Operations lead deprecates the Service Level Definition*

This Deprecated state indicates that all existing service level agreements have been moved onto the alternative, compatible Service Level Definitions.

## 14.6.1 Deprecating service version

When all Service Level Definitions for a specific service version have been deprecated, the service version must be deprecated. Deprecating the service version indicates to any user, who is searching WSRR for potential services to consume, that they must not recommend this service version as a potential dependency.



In the case of the Ship Goods service version, there is only a single Service Level Definition that we deprecated in the previous step. The Shared Services operations lead deprecates the service version, as shown in Figure 14-12 on page 323.

JKHLE

Service Registry and Repository

Perspective: Operations

Home

Actions

View

Tasks

My Service Registry

Help

Service Version

Service Level Definitions > SLD - Ship goods > Ship goods

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods

Description

Organise the shipment of goods

Version

2.0.1

Consumer Identifier

SSSG020

Version Availability Date

Monday, 27 July 2009

Version Termination Date

Monday, 27 July 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B1CCE6CC6-AF9D-61BA-2600-9A9692FE1B20%7D&v=2.0.1

Remote State

Approved

Owner Email

Additional Properties

Back

New Version

Deprecate

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

None

Provides

SLD - Ship goods

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

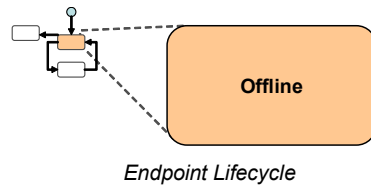
None

Provider DOU(s)

Figure 14-12 Shared Services operations lead deprecates the service version

Chapter 14. Service Level Definition management 323

## 14.7 Retiring a Service Level Definition



The endpoints associated with the Service Level Definition can now be revoked from use to stop any consumer from invoking them. The operations lead revokes both the staging and production endpoints that are associated with the Service Level Definition. Figure 14-13 on page 325 shows the operations lead revoking the production endpoint from use.



**Service Registry and Repository**

Perspective: Operations | jon | Log

Home | Actions | View | Tasks | My Service Registry | Help

### SOAP Service Endpoint

**Online Endpoints** > <http://production.jkhle.com:9080/governance/services/sg>

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

Details | Impact Analysis | Governance | Policy | Activity

[Edit Properties](#) | [Edit Relationships](#) | [Edit Classifications](#)

☒ **Properties**

**\*Name**

**Description**

**Namespace**

**Owner**

**Version**

**Last modified**

**\*Endpoint Type**

**\*Port Name**

**Service Version**

**\*Service Namespace**

**\*Service Name**

☒ **Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

☒ **Relationships**

**Soap Address**

**WSDL Ports**

**Source Document**

☒ **Governance State**

☒ **Classifications**

Production

SOAP Service Endpoint

Online

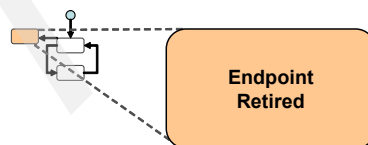
☒ **Additional Properties**

Back
Revoke From Use

**Help**

**Field help**  
For field help, information, field label or when the help appears.

Figure 14-13 Operations lead revokes the endpoints from use



Endpoint Lifecycle

With the endpoints out of service, the operations lead can then retire them to make them permanently unavailable, as shown in Figure 14-14 on page 326.

**Service Registry and Repository**

Perspective: Operations

[Home](#)
[Actions](#)
[View](#)
[Tasks](#)
[My Service Registry](#)
[Help](#)

### SOAP Service Endpoint

**Online Endpoints** > <http://production.jkhle.com:9080/governance/services/sg>

Detail view for SOAP Service Endpoint. A SOAP Service Endpoint represents the governed Service Endpoint for any services using SOAP addressing to define their endpoints.

[Details](#)
[Impact Analysis](#)
[Governance](#)
[Policy](#)
[Activity](#)

[Edit Properties](#)
[Edit Relationships](#)
[Edit Classifications](#)

**Properties**

- \*Name  
<http://production.jkhle.com:9080/governance/services/sg>
- Description
- Namespace  
<http://www.jkhle.com/governance/service/sg>
- Owner  
jon
- Version  
1.0
- Last modified  
Tuesday, 4 August 2009 18:56:12 ET
- \*Endpoint Type  
SOAPAddress
- \*Port Name  
ShipGoodsPort
- Service Version  
1.0
- \*Service Namespace  
<http://www.jkhle.com/governance/service/sg>
- \*Service Name  
ShipGoodsService-Production

**Links**

- Graphical View
- Applied Policies
- Applied Policy Attachments

**Relationships**

- Soap Address  
[ShipGoodsPort\\_SOAPAddress](#)
- WSDL Ports  
[ShipGoodsPort](#)
- Source Document  
[ShipGoodsProduction.wsdl](#)

**Governance State**

- Offline

**Classifications**

- Offline
- Production
- SOAP Service Endpoint

**Additional Properties**

[Back](#)
[Approve For Use](#)
[Retire From Use](#)

Figure 14-14 Operations lead retires the endpoint

With all endpoints revoked from use, the SOA Governance CoE representative can now complete the Service Level Definition lifecycle by retiring the Service Level Definition. Figure 14-15 on page 327 shows the CoE representative retiring the Service Level Definition.

JKHLE

Service Registry and Repository

Perspective: SOA Governance ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Extended Service Level Definition

Service Level Definitions > SLD - Ship goods

Detail view for Service Level Definition. The Service Level Definition provides additional quality of service information for a subscribable endpoint. This can be used to define particular metrics associated with endpoints.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

Name

SLD - Ship goods

Description

Specifies the level of service which will be provided as standard to the Web site

Average Response Time

4 seconds

Availability

24/7 High Availability

Additional Properties

Back

Retire

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Service Interface

ShipGoods

Available Endpoints

<http://production.jkhle.com:9080/governance/services/sg>  
<http://staging.jkhle.com:9080/governance/services/sg>

Bound SCA Export

None

Bound Web Service Port

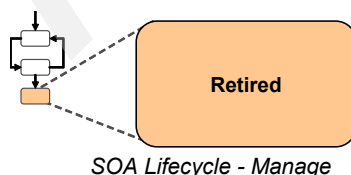
ShipGoodsPort

Figure 14-15 CoE representative retires the Service Level Definition

This step completes the governance process of migrating consumers to alternative providers. You can see from these steps how to complete the Service Level Definition lifecycle.

## 14.7.1 Retiring the service version

When all Service Level Definitions for a specific service version have been retired, the service version itself must be retired. Retiring the service version provides an indication to WSRR users that the service version is no longer available within any environment.



In the case of the Ship Goods service version, there is only a single Service Level Definition, which was retired in the previous step. The SOA Governance CoE representative retires the service version, as shown in Figure 14-16 on page 328.

JKHLE

Service Registry and Repository

Perspective: SOA Governance ▾

Home

Actions ▾

View ▾

Tasks ▾

My Service Registry ▾

Help ▾

Service Version

Service Versions > Ship goods

Detail view for Service Version. A Service Version represents a specific version (or release) of a Service and provides a range of functional and non functional specifications that hold for that version of the service. The Service Version exposes its capabilities as service level definitions. It may also (in the case of a composite service) identify the services it depends on by defining Service Level Agreements to the Service Level Definitions provided by the consumed service.

Details

Impact Analysis

Governance

Policy

Activity

Edit Properties

Edit Relationships

Edit Classifications

Properties

\*Name

Ship goods

Description

Organise the shipment of goods

Version

2.0.1

Consumer Identifier

SSSG020

Version Availability Date

Monday, 27 July 2009

Version Termination Date

Monday, 27 July 2009

Version Requirements Link

urn:serviceregistry

Asset Web Link

https://sys3.itso.ral.ibm.com:9443/ram/assetDetail/generalDetails.faces?guid=%7B1CCE6CC6-AF9D-618A-2600-9A9692FE1B20%7D&v=2.0.1

Remote State

Approved

Owner Email

Additional Properties

Back

New Version

Reject Deprecation

Retire

New SLD

Links

Graphical View

Applied Policies

Applied Policy Attachments

Relationships

Interface Specifications

None

Provided Web Services

None

Provided SCA Modules

None

Owning Organization

Shared services

Dependency

None

Aggregation

None

Artifacts

None

Provides

SLD - Ship goods

Consumes

None

Dependent Entities

Chartered Business Capability(s)

Ship goods

Consumer DOU(s)

None

Provider DOU(s)

DoU Clothing ordering -> Ship goods

Figure 14-16 CoE representative retires the service version

328 Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition

# Reporting

This chapter will examine the reporting requirements that JKHL Enterprises (JKHLE) has determined are important to help them assess the success and acceptance of their Service governance solution.

Specifically, JKHLE has chosen three reports:

- ▶ Business capability reuse
- ▶ Business capabilities by lifecycle state
- ▶ Business capabilities by organization

## 15.1 Introduction

WebSphere Service Registry and Repository (WSRR) has a reporting tool that is delivered as both an Eclipse plug-in that can be placed within any Eclipse environment and as an integrated part of the WSRR Studio tooling. You can obtain the full documentation for the reporting tool in the WSRR Information Center:

[http://publib.boulder.ibm.com/infocenter/sr/v6r3/topic/com.ibm.sr.studio.doc/cwsr\\_mansrvce\\_studio\\_reports.html](http://publib.boulder.ibm.com/infocenter/sr/v6r3/topic/com.ibm.sr.studio.doc/cwsr_mansrvce_studio_reports.html)

The WSRR reporting tool is based on a powerful open source reporting system called Business Intelligence and Reporting Tools (BIRT™). For more information, refer to the BIRT project Web site:

<http://www.eclipse.org/birt/phenix>

We develop the three reports in this chapter by using the WSRR reporting tool. We have made all three reports available for you to download and use. The download instructions are in Appendix B, “Additional material” on page 363.

## 15.2 WSRR report tool

We briefly describe the WSRR report tool and, additionally, the BIRT capabilities.

Choose the Report Design perspective when working with reports so that all the correct views are available for use. You must specify a project within which the reports can be placed. The project can be either a new project created specifically to contain the reports, or an existing project that contains other assets. A report can then be created within the project.

There are five concepts that are critical to understand when developing a new report:

- ▶ Data source
- ▶ Data set
- ▶ Joint data set
- ▶ Report table
- ▶ Report chart

We briefly explain these concepts. They are a guide to the capabilities that the WSRR report tool offers.

# 15.2.1 Data source

A report must specify a *data source*, which defines a connection to a specific WSRR server that will be queried to produce the report data.

There are two ways to set up the connection to the WSRR server depending upon your environment:

- ▶ When using the Eclipse plug-in within an Eclipse environment, the connection is fully specified within the Data Source section of a report.
- ▶ When using the WSRR studio tool, connections to WSRR servers are defined at the workspace level and then reused within the data source of the report.

Figure 15-1 shows setting up the connection to the WSRR server within the WSRR Studio preferences panel.

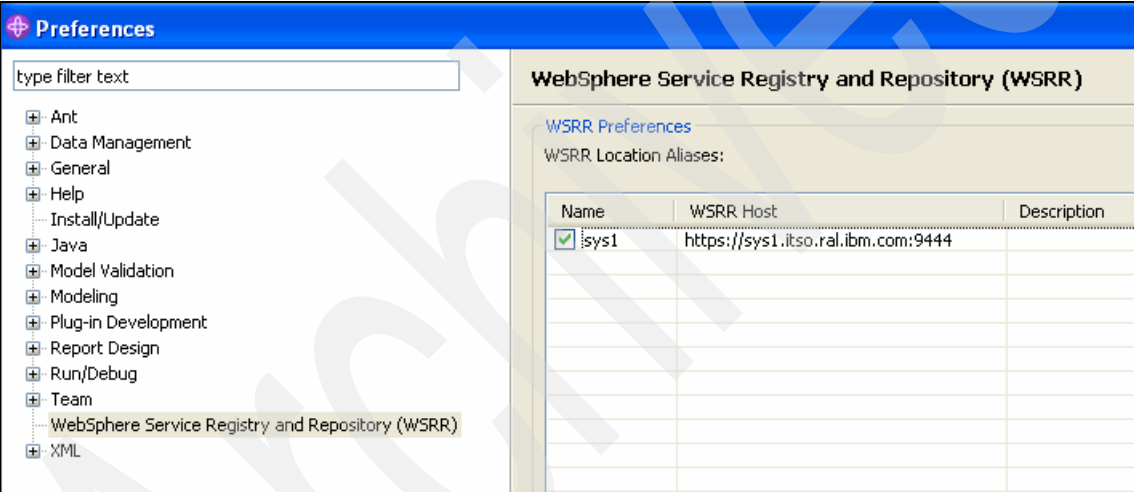


Figure 15-1 Configuring WSRR server connections

Figure 15-2 on page 332 shows setting up the data source of a report by reusing the connection details that are already defined.

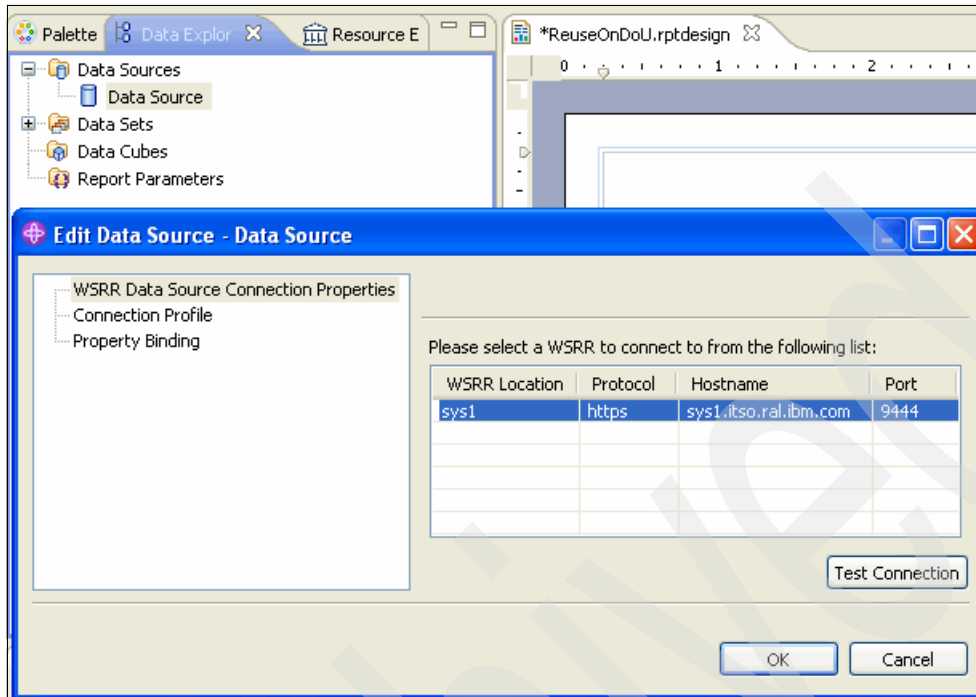


Figure 15-2 Referencing the WSRR connection in a report data source

## 15.2.2 Data set

A *data set* represents a request for data from the WSRR server. The request for data takes the form of an XPath query. The data set can specify that additional properties, relationship targets, and classifications are returned by the query. Figure 15-3 on page 333 shows the creation of a data set that queries for all Business Capability entities and additionally returns all relationship targets and classifications.



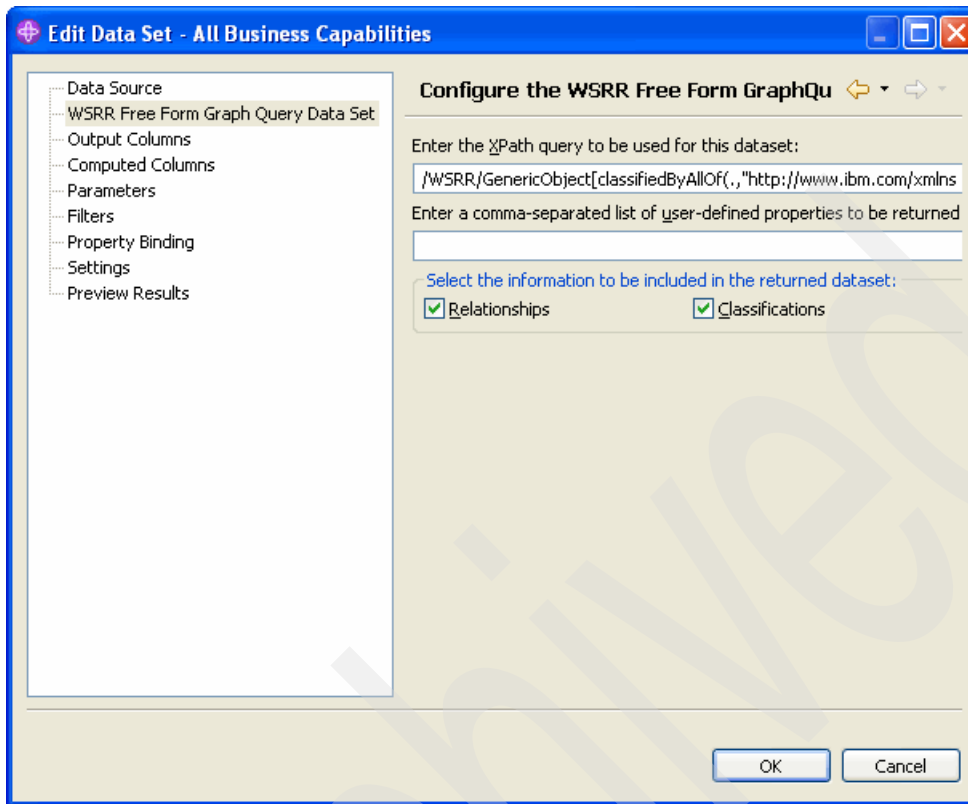


Figure 15-3 Creating a data set to retrieve all Business Capabilities

Figure 15-4 on page 334 shows a preview of the results for the query, including the relationship targets and classifications.

name	version	relationship name	targetBsrURI	classification URI
Ship goods	1			
Ship goods	1	gep63_capabilityVe...	3cd8413c-876...	
Ship goods	1	gep63_capabilityVe...	fbdc65fb-1625...	
Ship goods	1	ale63_owningOrgan...	736ccb73-e14...	
Ship goods	1	ale63_artifacts	667ac766-733...	
Ship goods	1	gep63_charter	667ac766-733...	
Ship goods	1			http://www.ibm.co...
Ship goods	1			http://www.ibm.co...
Ordering Application	2			
Ordering Application	2	gep63_capabilityVe...	c22ab7c2-af04...	
Ordering Application	2	ale63_owningOrgan...	80933b80-bd5...	
Ordering Application	2	ale63_artifacts	c8f51bc8-2075...	
Ordering Application	2	gep63_charter	c8f51bc8-2075...	
Ordering Application	2			http://www.ibm.co...
Ordering Application	2			http://www.ibm.co...
Clothing ordering	1.3			
Clothing ordering	1.3	ale63_owningOrgan...	13c96113-0f26...	
Clothing ordering	1.3	ale63_artifacts	468bbd46-437...	
Clothing ordering	1.3	gep63_charter	468bbd46-437...	
Clothing ordering	1.3			http://www.ibm.co...
Clothing ordering	1.3			http://www.ibm.co...

Figure 15-4 A preview of the data set results

The results are returned in a tabular format, with rows for each result and columns for the various within the results.

Note that each result object of the XPath query is returned as a row, with additional rows for every target of each relationship and for every classification of the result object.

So, a Business Capability with one classification and with a relationship “re1Name” that has two target objects will return four rows:

- ▶ The Business Capability with no relationship or classification
- ▶ The Business Capability with no relationship but with the classification
- ▶ The Business Capability with the relationship “re1Name” and the details of the first target object, but with no classification
- ▶ The Business Capability with the relationship “re1Name” and the details of the second target object, but with no classification

The data set also has filtering functionality that is based on the data returned. Therefore, in the previous results, it is possible to create a filter specifying that the relationship name is not null. This specification filters out the first two results and leaves only the latter two results. Figure 15-5 on page 335 shows the addition of the relationship name is not null filter.

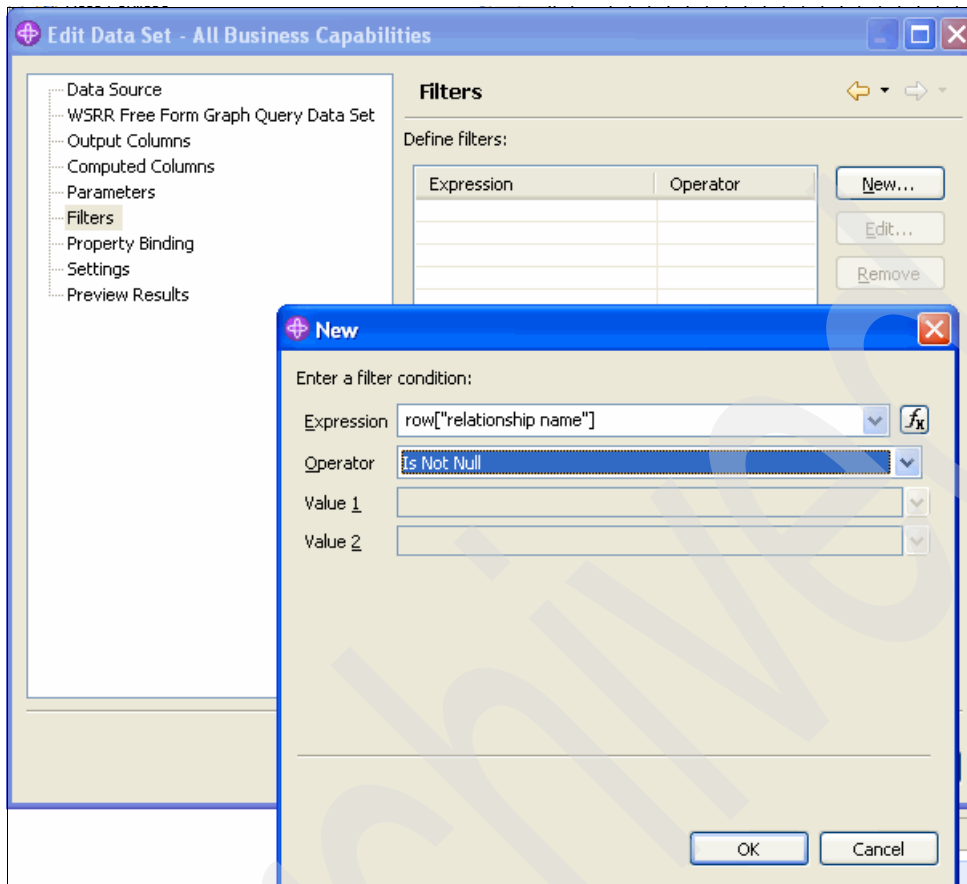


Figure 15-5 Adding a filter to the data set specifying that the relationship name is not null

### 15.2.3 Joint data set

You can perform joins on the results of data sets. These joins are similar to joins that are used within SQL statements and can take the form of inner joins, left outer joins, right outer joins, or full outer joins.

Briefly, there are four join types:

- Inner join** Only return rows where the join matches in both data sets.
- Left outer join** Return all rows from the left data set even when there are not matching rows in the right data set.
- Right outer join** Return all rows from the right data set even when there are not matching rows in the left data set.

### Full outer join

Return all rows from both the left and right data sets regardless of whether there are matches.

Figure 15-6 shows the creation of a *joint data set* between two existing data sets.

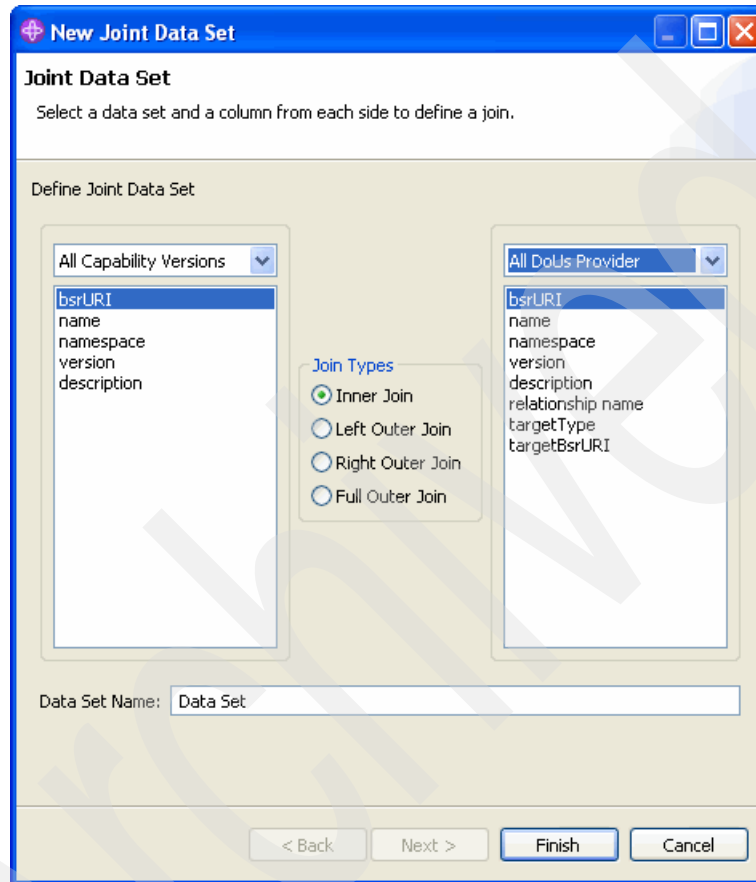


Figure 15-6 Creating a joint data set

Joint data set is an extremely powerful feature of the reporting, because it allows the report writer to develop complex result sets that simple XPath queries do not allow.

We use Joint data sets extensively within the reports that are designed within this chapter.

## 15.2.4 Report table

After the data sets have been defined, you can create a *report table* to show the results of a specific data set in a tabular format.

The report tables consist of a header row, a data row, and a footer row. The table is associated to a data set, and then, you can assign columns from the data set's results to cells within the data row of the table. When the report is generated, the data row of the table is expanded to multiple rows, one row for each of the returned results.

Additionally, you can create result groups. This function allows you to group result column values to improve the readability of the report.

We use report tables extensively within the reports that are designed within this chapter.

## 15.2.5 Report chart

After the data sets have been defined, you can create a *report chart* to show the results of a specific data set in chart format. Many chart types are available, including bar charts and pie charts, which will be used as part of the reports that are designed within this chapter.

The Report Chart Properties panel allows the assignment of various result columns to separate axes or series. Additionally, functions can be enacted on the result columns. For example, the count function counts how many times a results repeats so that the count can be used in a bar chart or pie chart.

## 15.3 Business Capability reuse

JKHLE wants to ensure that the Service governance solution that is in use fosters an increase in the level of reuse within the company. Measuring reuse will give JKHLE a metric to use to report the financial savings that are achieved through the new service governance strategy. Clearly, increasing the reuse of capabilities reduces the expenditure on development and production due to a reduction in duplication. This duplication can be in the form of more development team members to create the capabilities, more hardware to run the capabilities, and more operations team members to deploy and manage the capabilities.

The JKHLE reuse report will form two parts:

- ▶ A Bar chart showing a high-level measure of Business Capability reuse
- ▶ A Table showing a complete breakdown of which Capability Versions are reused by which other Capability Versions

Both reports are based on the Documents of Understanding (DOUs) that have been put in place between the provider and the consumer Capability Versions. JKHLE has chosen to use the DOU relationships as the basis of its reporting, because it wants to get a complete picture of reuse from the modeling stage through the development stages to the production stage of Capability Versions.

**Note:** The examples of reporting reuse focus on the DOU relationship. However, you can also apply the concepts for creating reports that are outlined in this section to other relationships, such as the Service Level Agreement (SLA) to Service Level Definition (SLD) relationship.

### 15.3.1 Bar chart showing Business Capability reuse

Figure 15-7 shows the structure of the data on which JKHLE wants to report.

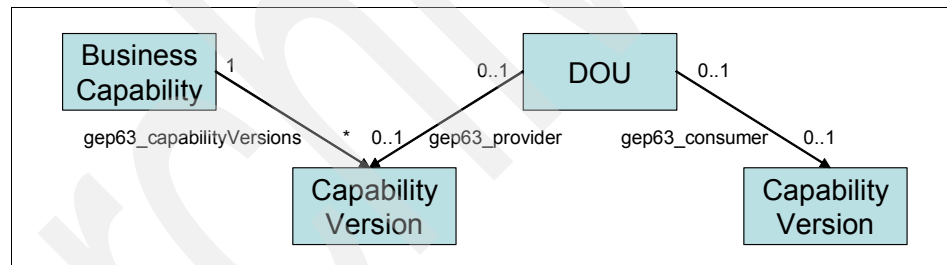


Figure 15-7 The data structure for the reporting

This data structure is a slightly complicated structure, and therefore, we need to create several data sets and joint data sets to allow reporting on this data.

#### All Business Capabilities data set

A data set is created to return all the Business Capabilities, and their relationships, that are stored within the WSRR server that is specified within the data source for the report.

Additionally, because the report focuses on the reuse of Business Capabilities, and in particular Capability Versions, the results are filtered on the relationship name equal to “gep63\_capabilityVersions.”

Table 15-1 shows an overview of the settings for the data set.

Table 15-1 Attributes of the data set

Attribute	Value
Name	AllCaps
XPath query	/WSRR/GenericObject[classifiedByAllOf(., "http://www.ibm.com/xmlns/prod/serviceregistry/profile/v6r3/GovernanceEnablementModel#BusinessCapability")]
Relationships	Selected
Classifications	Not selected
Filters	Property relationship name equal to "gep63_capabilityVersions"

The result set therefore contains a row for each target of a relationship named "gep63\_capabilityVersions" from a Business Capability. Business Capabilities might appear multiple times if they have multiple targets on their "gep63\_capabilityVersions" relationship, and also, Business Capabilities that have no targets on their relationship will not appear.

**All DOUs data set**

A data set is created to return all the DOUs, and their relationships, that are stored within the WSRR server.

The results are filtered on the relationship name equal to "gep63\_provider". The result set therefore contains a row for each target of a relationship named "gep63\_provider" from a DOU. DOUs will either appear in the result list if they have a provider specified, or they will not appear in the result list.

Table 15-2 on page 340 shows an overview of the settings for the data set.

Table 15-2 Attributes of the data set

Attribute	Value
Name	DoUProvider
XPath query	/WSRR/GenericObject[classifiedByAllOf(., "http://www.ibm.com/xmlns/prod/serviceregistry/profile/v6r3/GovernanceEnablementModel#DoU")]
Relationships	Selected
Classifications	Not selected
Filters	Property relationship name equal to "gep63_provider"

**Note:** For simplicity in this report, there is no checking that the DOU has a consumer. We assume that a DOU with a provider also has a consumer. Another report will show how to create more advanced reporting.

## Business Capability to DOU joint data set

Finally, the two data sets just defined are joined to create the data set that can be used for the bar chart.

The "All capabilities" data set is joined on its targetBsrURI property to the "All DOUs" data set on its targetBsrURI property. The join used is a left outer join so that results from the "All Capabilities" data set remain in the result list even if there are no joining rows from the "All DOUs" data set. Or more relevantly, Business Capabilities will remain in the results even if they do not have any Capability Versions that are reused.

Table 15-3 shows an overview of the settings for the joint data set.

Table 15-3 Attributes of the data set

Attribute	Value
Name	DoUToCap
Left join	AllCaps on targetBsrURI property
Right join	DoUProvider on targetBsrURI property
Join type	Left outer join
Filters	None



Therefore, the result set will return all Business Capabilities that have one or more Capability Versions that are providers for DOUs. This result is equivalent to all Business Capabilities that have Capability Versions that are reused by another Capability Version and therefore gives the amount of reuse within the company.

### **Bar chart showing reuse**

We add a chart object to the report and select the 2D bar chart type. Figure 15-8 on page 342 shows the selection of the charting type.

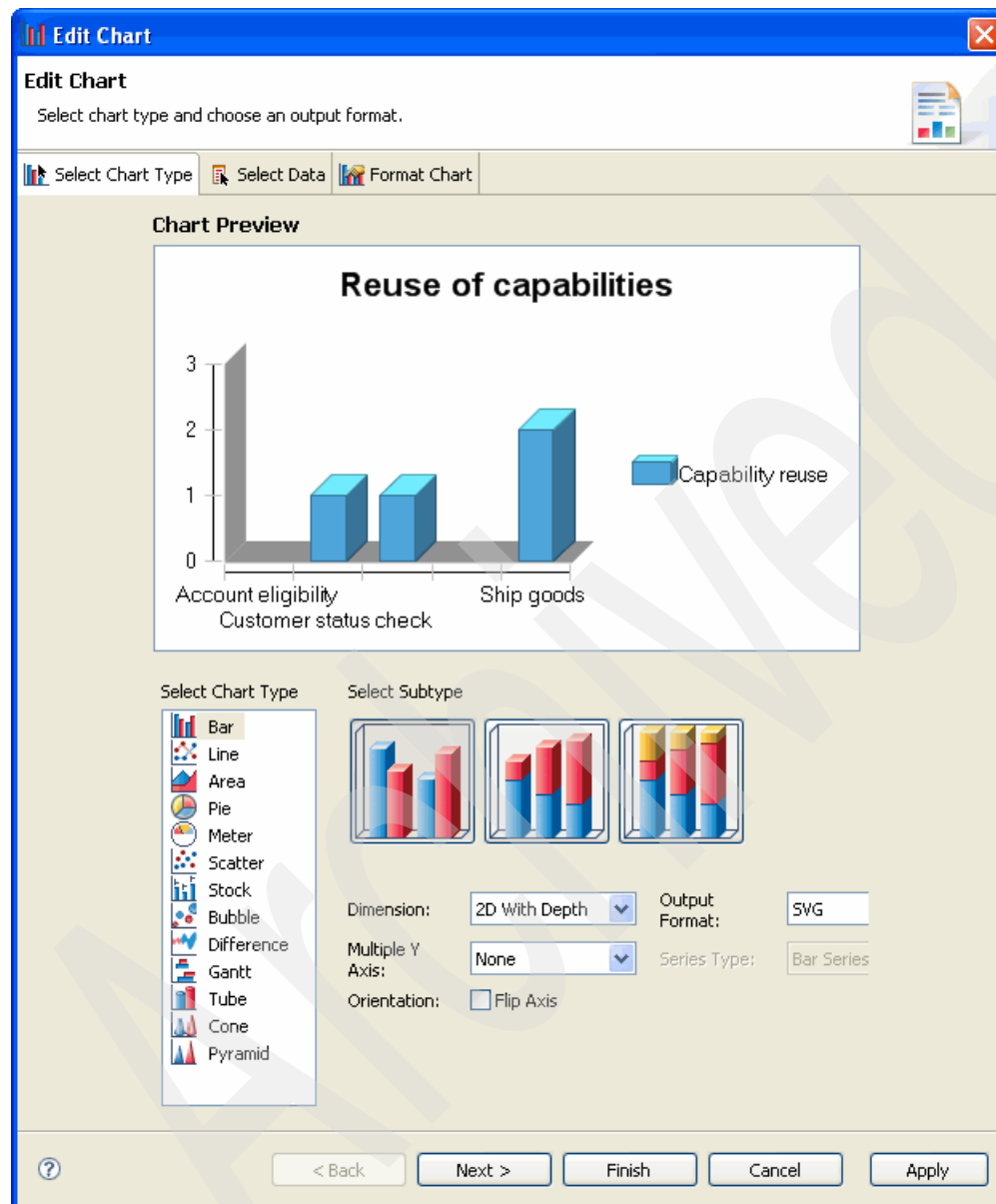


Figure 15-8 Creating a report chart by first selecting the chart type

The data to be used is set to the data set of “DoU to Business Capability.” The value of the X axis is set to the name of the Business Capability, which is sorted and grouped. The value of the Y axis is set to the name of the DOU, and the aggregation is set to the “count” option so that the Y axis shows the count of

DOUs for each Business Capability. Figure 15-9 shows the specification of axis data.

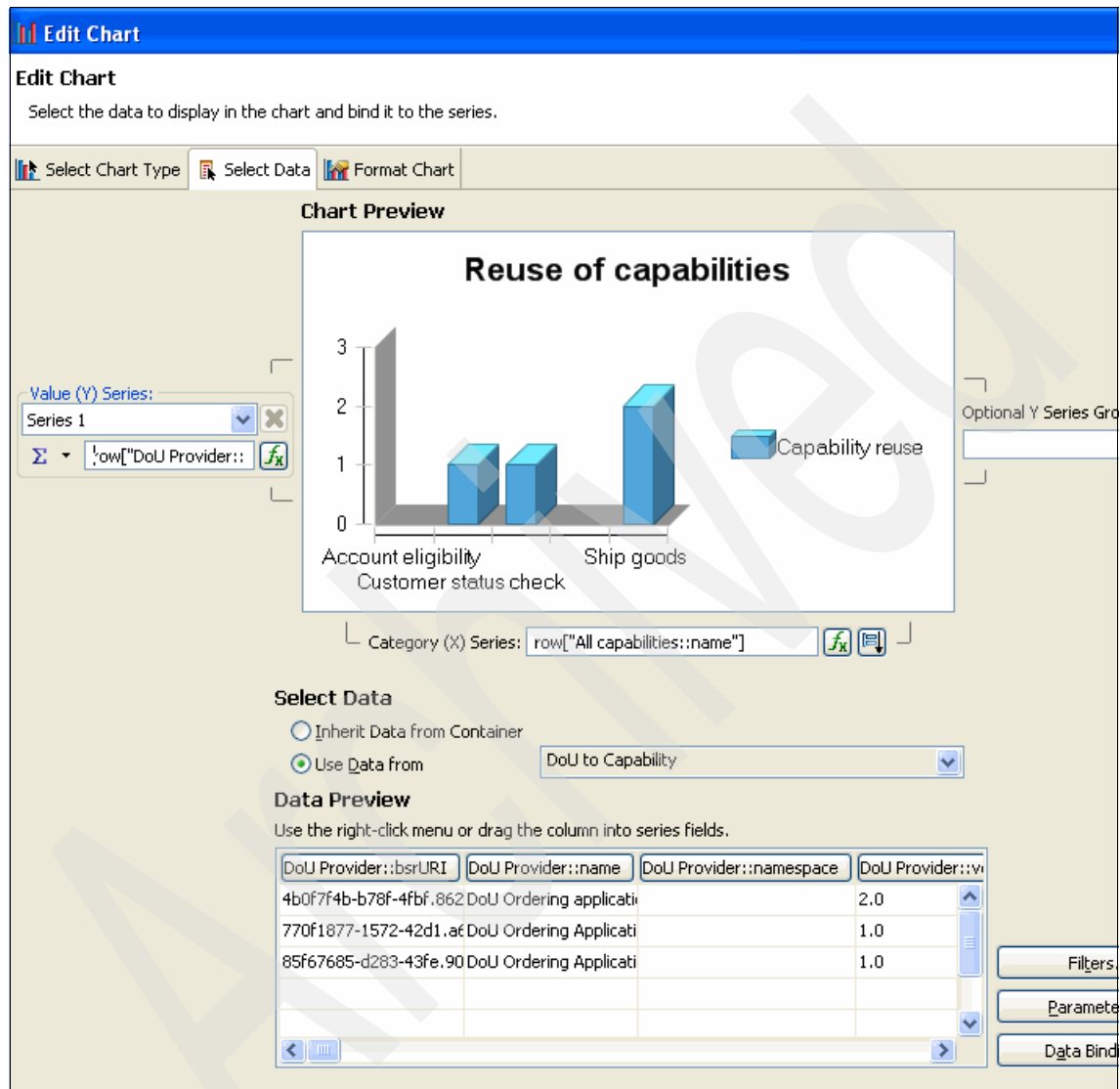


Figure 15-9 Specifying the data binding for the chart

Finally, we specify the names of the axis and the chart so that the chart is self-explanatory. Figure 15-10 on page 344 shows setting the names for the chart.

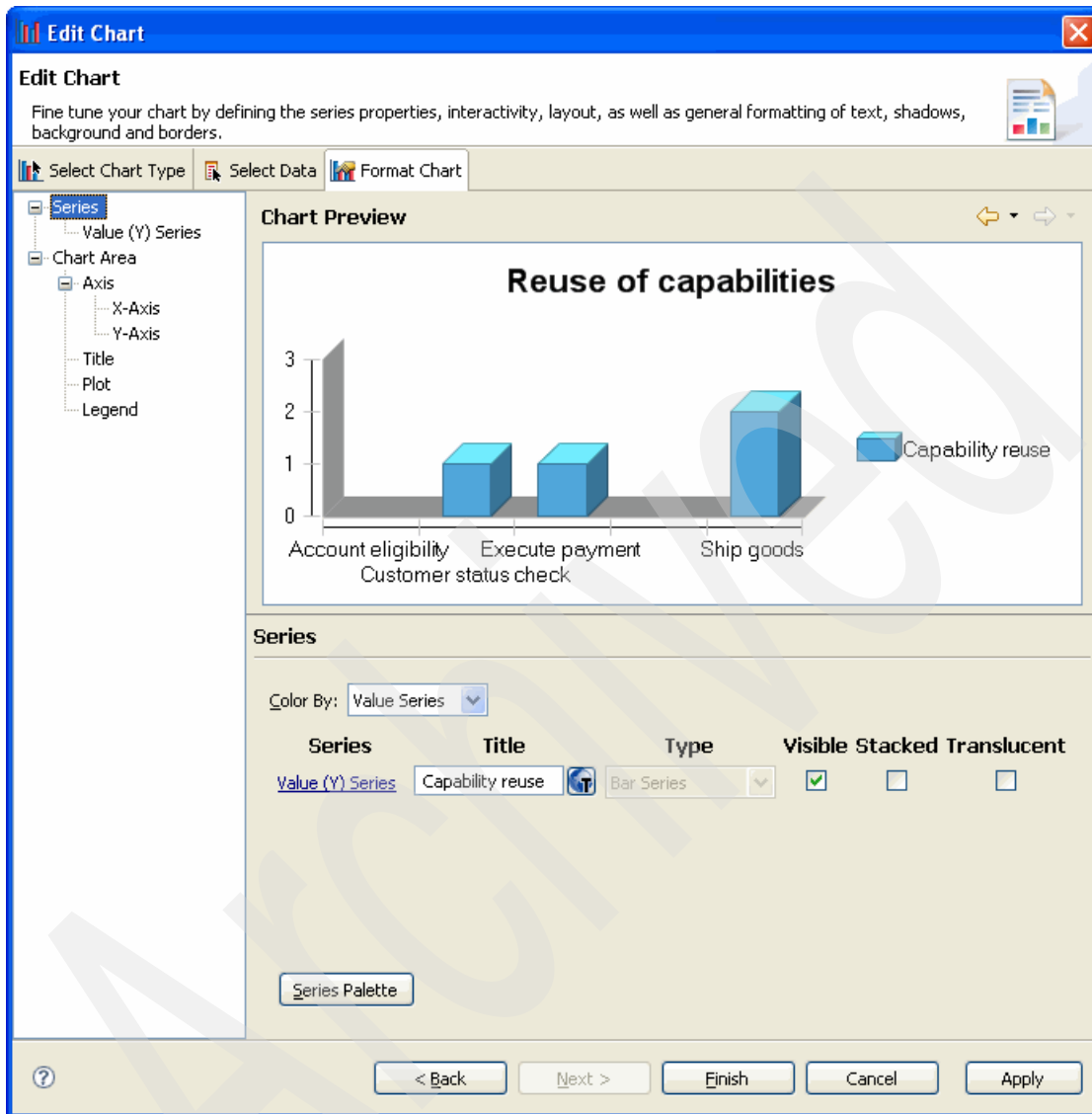


Figure 15-10 Specifying the titles for the chart

We have set up the chart, which is shown on the report design page. Now, we can run the report and put the output in a series of formats, including HTML and PDF. Figure 15-11 on page 345 shows a generated report.

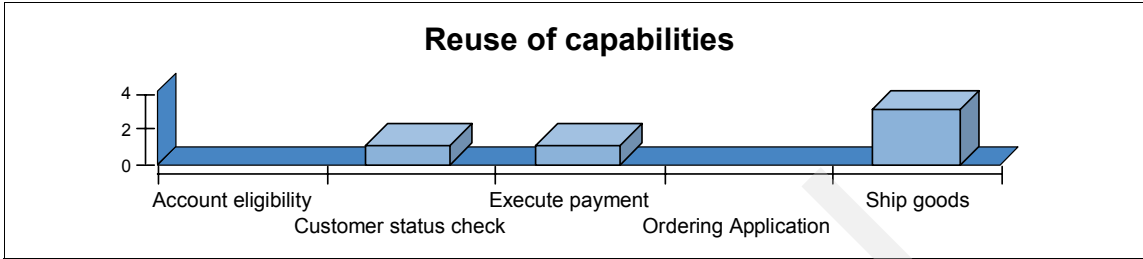


Figure 15-11 A generated report showing the reuse of Business Capabilities

### 15.3.2 Table showing Business Capability reuse

We need a tabulated form of the data that is displayed in the chart so that we can determine the actual Capability Versions that are reused and by what.

Many data sets are involved:

<b>BusCap</b>	All Business Capabilities with relationships, filtered on the relationship name “gep63_capabilityVersions”
<b>CapVer</b>	All Capability Versions
<b>BusCapToCapVer</b>	An inner join of the “BusCap” data set targetBsrURI property and the “CapVer” data set bsrURI property to give all of the Business Capabilities and the details of the Capability Versions to which they relate

**Note:** We used an inner join here (unlike in the report before), because we do not want to keep Business Capabilities that do not have Capability Versions. We continue to use inner joins in this report, because we want the report to only show reuse. Therefore, Business Capabilities without Capability Versions, Capability Versions that are not DOU providers, and DOUs that do not have a consumer are not important to the report.

<b>DoUProvider</b>	All DOUs with relationships, filtered on the relationship name “gep63_provider”
<b>DoUConsumer</b>	All DOUs with relationships, filtered on the relationship name “gep63_consumer”
<b>DoUProvToCons</b>	An inner join of the “DoUProvider” data set bsrURI property and “DoUConsumer” data set bsrURI property to give all DOUs with both the provider and the consumer relationship in the same row

**Note:** It might seem simpler to create a single data set that is filtered on both the “gep63\_provider” and “gep63\_consumer” relationships. However, doing so creates a result set with the same DOU on two separate rows, one DOU with the provider relationship and one DOU with the consumer relationship. As shown next, we require that the DOU is only on one row with the details of both relationships.

- BusCapToDoU

An inner join of the “BusCapToCapVer” data set Capability Version bsrURI property and the “DoUProvToCons” data set provider targetBsrURI property. This inner join returns the Business Capability, provider Capability Version, and DOU all on the same row.
- Reuse

An inner join of the “BusCapToDoU” data set DOU consumer targetBsrURI property and the “CapVer” data set bsrURI property. This inner join returns the Business Capability, provider Capability Version, DOU, and consumer Capability Version all on the same row.

The “Reuse” data set is linked to the report table to give full details of the reuse of Business Capabilities within the company.

The header labels for the table are created in the header row. Then, a group is created to group all the Business Capabilities on the name property, which ensures that all reuse of a Business Capability is in contiguous rows. A subgroup is then created to group the Capability Versions on the version property, which ensures that all reuse of the Capability Versions is in contiguous rows. Finally, the data rows are populated with the providing Capability Version name and version properties. Figure 15-12 shows the specification of the report table.

	Business capability	Capability Version	Reused capability version	
1	[Cap to Version t...]			
2	ion to DoU; Capability To Version	[Cap to Version t...]		
	Detail Row		[Column Binding]	[Column Binding_1]
	Table			

Figure 15-12 Specification of the contents of the report table

Additional formatting is added to the table to highlight the header row and to add lines to demarcate the rows and columns of the table. Also, the two group footer rows and the table footer row are removed.

### 15.3.3 Report of Business Capability reuse

The report is then generated to give the reuse metrics for JKHLE as shown in Figure 15-13.

Business Capability	Capability Version	Reused capability version	
Customer status check			
	1.0.0		
		Ordering application version 2.0	2.0
Execute payment			
	1.0.3		
		Ordering application version 2.0	2.0
Ship goods			
	2.0.1		
		Clothing ordering	1.1
		Footwear ordering	1.3
	2.1		
		Ordering application version 2.0	2.0

Figure 15-13 Generated report of Business Capability reuse

## 15.4 Business Capabilities by lifecycle state

JKHLE wants to see the ramping up of Business Capabilities within their Service Governance repository. JKHLE wants to report on how many Business Capabilities are at each state within their lifecycle. In this way, JKHLE ideally can see the number of approved Business Capabilities increase and also see the new Business Capabilities passing from the Identified through the Charter Review to the Approved state. JKHLE needs to monitor the rate at which new Business Capabilities are added to the repository.

JKHLE wants a high-level pie chart that shows Business Capabilities by lifecycle state and a table describing which Business Capabilities are in which lifecycle states.

### 15.4.1 Pie chart for Business Capabilities by lifecycle state

The pie chart only requires the definition of one data set. This data set is a query for all Business Capabilities, with classifications selected for the Business

Capabilities. We added an additional filter to stipulate that the classification property value must be one of the three lifecycle states through which a Business Capability can pass.

**Note:** The lifecycle state of an object in WSRR is exposed in two ways:

- ▶ The object is related to a Governed Collection object, which has a state.
- ▶ The classification of the lifecycle state is added to the classifications of the object.

When using XPath queries, you can only use the latter way to query on lifecycle state. Therefore, the data set uses a filter on classifications to determine in which lifecycle states the Business Capabilities exist.

With the filter in place, there is only one entry for each Business Capability, because a Business Capability can only be in one lifecycle state.

We created a report chart and selected the 2D pie chart type. We specified the data set that was defined as the data source for the chart. We set the category definition for the pie chart to the classification property, and we set the slice size definition to the count of each value of the classification property.

The chart shows the numbers of Business Capabilities in each lifecycle state. However, the classification for each lifecycle state is verbose, because it is a full uniform resource indicator (URI). Therefore, we manipulate the data binding to improve the readability of the pie chart. We modify the data binding for the pie chart by changing the mapping from classification URI to the value that is shown in Example 15-1.

*Example 15-1 Data binding modification made to improve readability of lifecycle state*

---

```
dataSetRow["classification URI"].substring(dataSetRow["classification URI"].indexOf('#') + 1)
```

---

The modification takes a substring of the classification URI property value from the point one character after the # is present in the classification URI property value up to the end of the classification URI property value.

Figure 15-14 on page 349 shows the Data Binding page with the modification to the classification URI mapping.



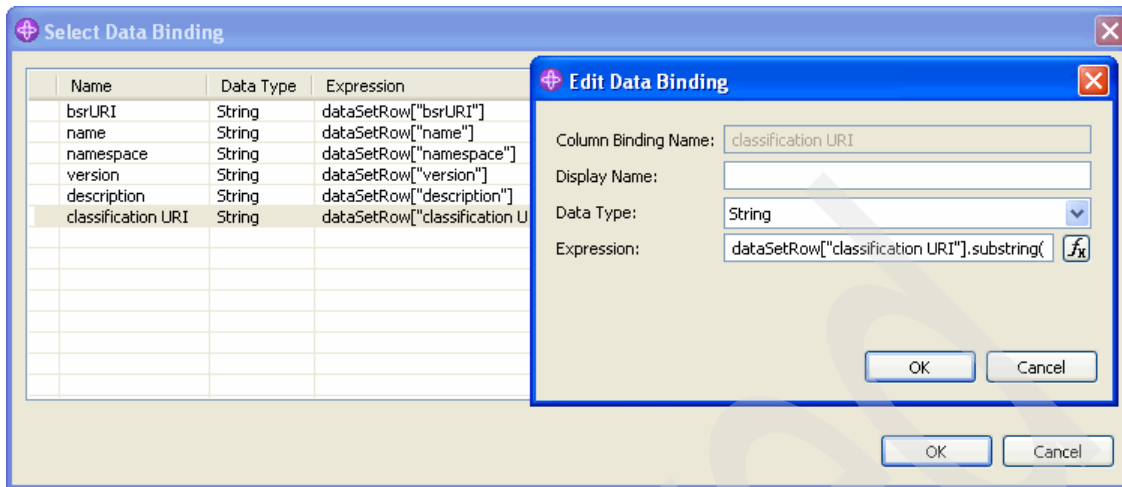


Figure 15-14 Modifying the data binding for the classification URI property

Finally, we specify the title of the pie chart and the category, completing the creation process.

## 15.4.2 Table for Business Capabilities by lifecycle state

We use the same data set that was created for the pie chart for the table.

We create a report table and bind the data set. We add heading labels to the table for “Lifecycle state” and “Business Capability.” We create a group for the classification URI property so that all Business Capabilities in a specific lifecycle state are grouped together. We then add the Business Capability name in the data row of the table.

The same problem with the lifecycle classification being too verbose also impacts the readability of the table. Therefore, to improve the readability of the table, we modify the data binding expression for the classification URI group, as shown in Example 15-1 on page 348.

Figure 15-15 on page 350 shows the Data Binding window containing the new binding expression.

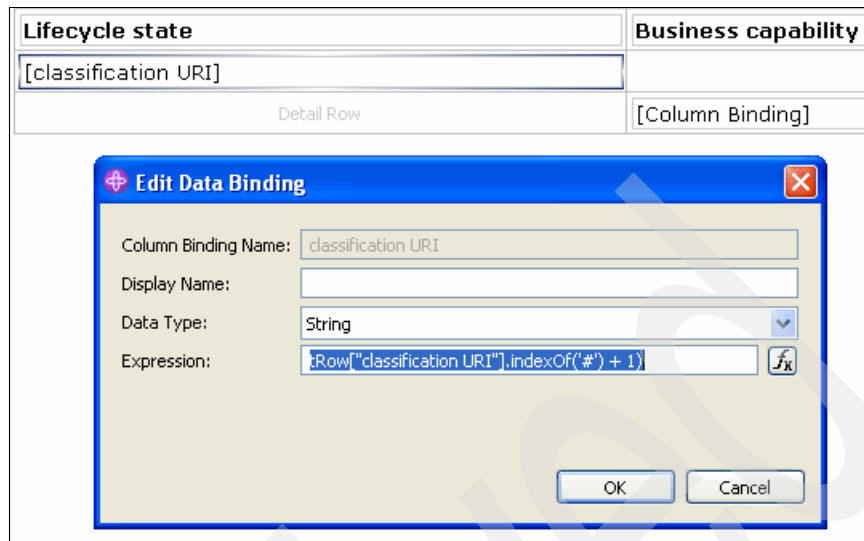


Figure 15-15 Specifying the data binding for the Classification URI property

We add additional formatting to the table to highlight the header row and to add lines to demarcate the rows and the columns of the table. Also, we remove the table footer row.

### 15.4.3 Report on Business Capabilities by lifecycle state

We generate this report to give the full reporting on Business Capabilities by lifecycle state as shown in Figure 15-16 on page 351.

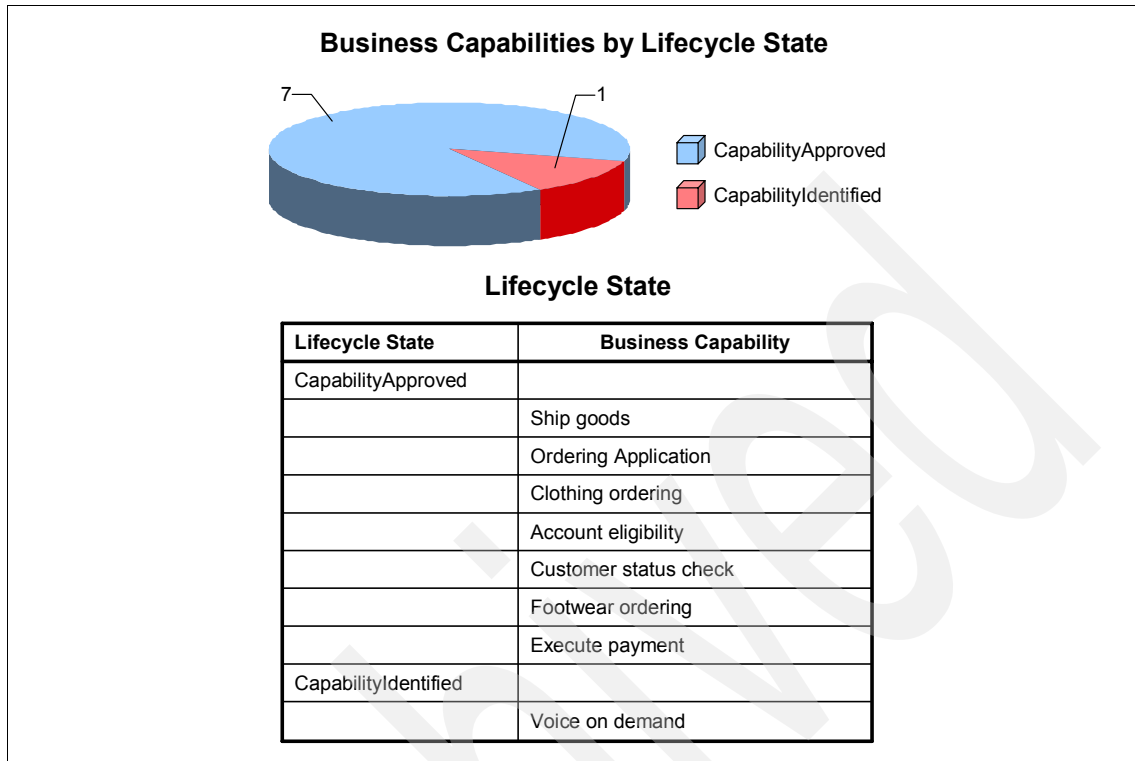


Figure 15-16 Report generated to show Business Capabilities by lifecycle state

## 15.5 Business Capabilities by Organization

The final report that is required by JKHLE shows which Organizations own which Business Capabilities. JKHLE wants to use this information to ensure that all parts of the JKHLE company use the Service Governance solution to govern their Business Capabilities.

JKHLE wants three parts in this report:

- ▶ A high-level bar chart showing the count of Business Capabilities by Organization
- ▶ A table showing the details of which Business Capabilities are owned by which Organizations
- ▶ A table showing Business Capabilities that do not have an owning organization, which are known as “rogue” Business Capabilities to JKHLE.

### 15.5.1 Bar chart for Business Capabilities by Organization

To show the number of Business Capabilities that are owned by each Organization, we need to define several data sets:

<b>JKHLE</b>	All Organizations with relationships, filtered on name property value “JKHLE” and relationship name property “ale63_child0rganization.” This data set returns the JKHLE Organization with relationships to all the suborganizations.
<b>AllOrgs</b>	All Organizations
<b>AllSubOrgs</b>	An inner join between the “JKHLE” data set targetBsrURI property and the “AllOrgs” data set bsrURI property. This data set returns all the Organizations one level beneath the JKHLE Organization.
<b>AllOwnedCaps</b>	All Business Capabilities with relationships, filtered on relationship name “ale63_owning0rganization”.
<b>OrgToCap</b>	A left outer join between the “AllSubOrgs” data set AllOrgs::bsrURI property and the “AllOwnedCaps” targetBsrURI property. We chose a left outer join, because all Organizations need to be reported, regardless of whether they own Business Capabilities. This data set returns all Organizations one level beneath the JKHLE Organization and an owned Business Capability on the same row.

Now, we can base the bar chart upon the “OrgToCap” data set, because it holds the details of all the suborganizations of the JKHLE organization and their relationship through the owning0rganization relationship to Business Capabilities.

We create a bar chart with the data set bound to the “OrgToCap” data set, the x axis bound to the Organization name property, and the y axis bound to the count of the Business Capability bsrURI property.

### 15.5.2 Table for Business Capabilities by Organization

The report table reuses the “OrgToCap” data set, because it already contains all the details that are required for the table.

We add a report table to the report, and we add heading labels to the table for “Organization” and “Business Capability.”

We create a group in the table for the Organization name property so that all Business Capabilities for each Organization are grouped together. The Business Capability name is added to the data row.

We then formatted the table in a similar style to the previous tables.

### 15.5.3 Table for Rogue Business Capabilities

We defined these data sets for this table:

<b>AllCaps</b>	All Business Capabilities
<b>AllCapsWithOrg</b>	All Business Capabilities that have an owning organization. This data set uses XPath to ensure that the Business Capability has a target on the “ale63_owningOrganization” relationship as shown in Example 15-2.

*Example 15-2 XPath query to retrieve all Business Capabilities with an owning organization relationship target*

---

```
/WSRR/GenericObject[classifiedByAnyOf(., 'http://www.ibm.com/xmlns/prod/serviceregistry/profile/v6r3/GovernanceEnablementModel#BusinessCapability')/ale63_owningOrganization(.)/@name]
```

---

<b>RogueCaps</b>	A left outer join between the “AllCaps” data set bsrURI property and the “AllCapsWithOrg” data set bsrURI property, filtered on the “AllCapsWithOrg” bsrURI property equal to null. This left outer join then removes the result rows that were in both the “AllCaps” and “AllCapsWithOrg” results, leaving just Business Capabilities that have no owning organization.
------------------	--

The rogue Business Capabilities report table is created and bound to the “RogueCaps” data set. A heading label “Rogue Business Capabilities” is added to the header row, and the Business Capability name property is added to the data row.

We format the table in a similar style to the other tables, and we remove the footer row.

### 15.5.4 Report on Business Capabilities by Organization

We generate the report to show all of the Business Capabilities by Organization as shown in Figure 15-17 on page 354.

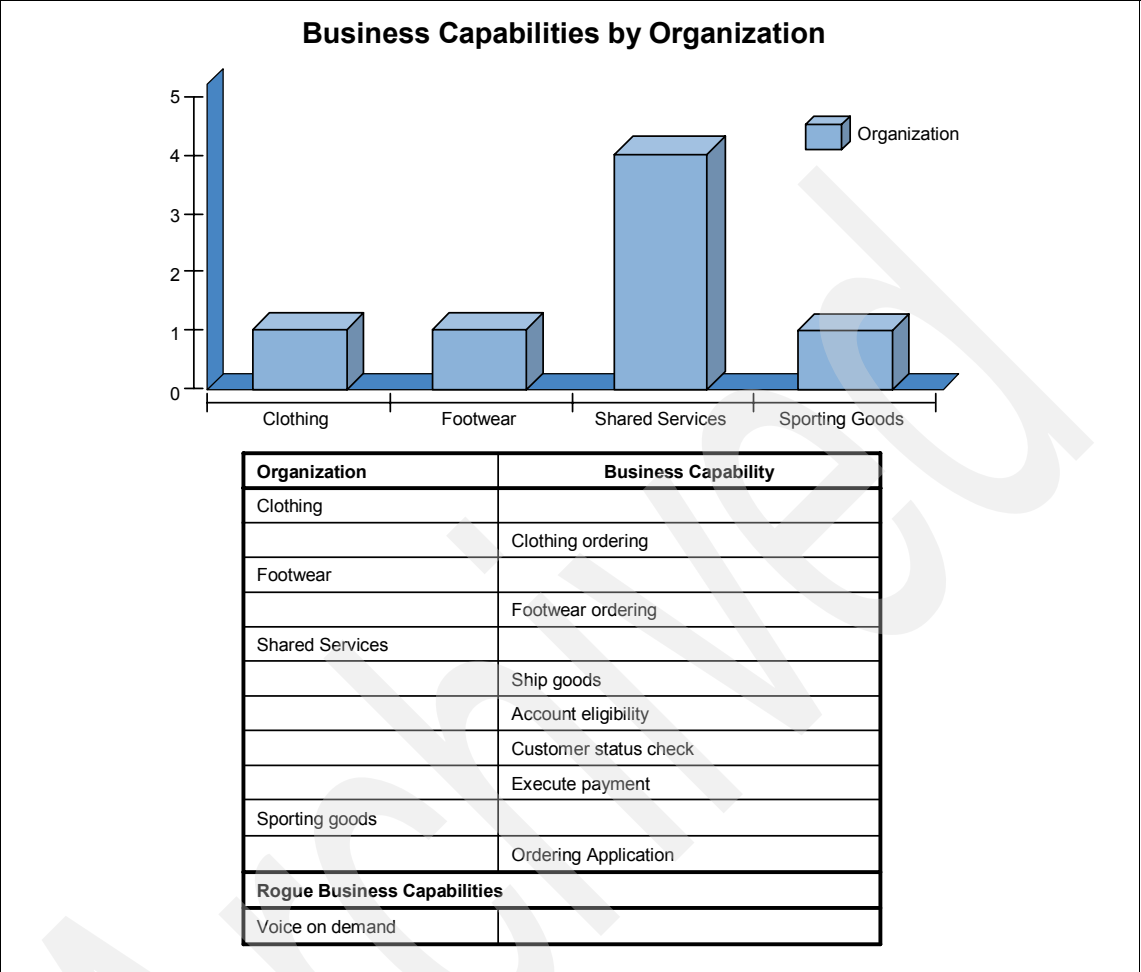


Figure 15-17 Generated report showing Business Capabilities by Organization



# Part 4

## Appendixes





## Configuring the tooling used in the scenarios

This appendix provides details about the actual configuration of the WebSphere Service Registry and Repository (WSRR) Advanced Lifecycle Edition tools. We used these tools to create the scenarios that were described in Part 2, “Top-down scenario” on page 95 and Part 3, “Other scenarios” on page 273 of this book.

We explain aspects of the configuration that are not discussed within the scenarios and details about those aspects of the configuration that are explored at a higher level within the scenarios.

Specifically, this appendix includes these topics:

- ▶ “Software versions and topology” on page 358
- ▶ “Security considerations” on page 358
- ▶ “Configuration” on page 359
- ▶ “Theme customization” on page 360
- ▶ “Charter validator” on page 361
- ▶ “Policy customization” on page 361

## Software versions and topology

We used the following runtime configuration to test the scenarios in this book:

- ▶ WebSphere Service Registry and Repository:
  - We used Fix Pack 1 of the V6.3 release (V6.3.0.1) in the scenarios. Fix Pack 1 includes a range of fixes made to improve the synchronization with Rational Asset Manager. Without this fix pack, many of the review steps in Rational Asset Manager require manual transitions of the correlated entity in WSRR.
  - WSRR was deployed to an instance of WebSphere Application Server V6.1.0.23 with application security enabled.
  - A local instance of DB2 was used to hold the WSRR database.
- ▶ Rational Asset Manager:
  - We used Rational Asset Manager V7.1.1.1 in the scenarios. Although WSRR Advanced Lifecycle Edition contains V7.1 of Rational Asset Manager, the synchronization between WSRR and Rational Asset Manager is only supported when using V7.1.1 or later.
  - Rational Asset Manager was deployed to an instance of WebSphere Application Server V6.1.0.23 with application security enabled.
  - A local instance of DB2 was used to hold the Rational Asset Manager database.

## Security considerations

The WebSphere Application Server instances for WSRR and Rational Asset Manager were disparate, and therefore, we were required to share the certificates between the two instances to ensure that secure communications were possible.

Additionally, we were required to share Lightweight Third-Party Authentication (LTPA) keys between the application servers. Exchanging these LTPA keys allows user credentials to be encrypted and decrypted between the application servers. The WebSphere Application Server Information Center explains the process that is undertaken to exchange these LTPA keys:

[http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec\\_sslmanage1ptakeys.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_sslmanage1ptakeys.html)

# Configuration

The JKHLE example used a specific set of roles and communities in the examples. This section outlines the usage of these users, roles, and communities.

## WebSphere Service Registry and Repository user to role mappings

In WSRR, we used the following user to role mappings:

- ▶ Development team in Development role and WSRRUser role
- ▶ Operations team in Operations role and WSRRUser role
- ▶ Service-oriented architecture (SOA) Governance Center of Excellence (CoE) in SOAGovernance role and WSRRUser role
- ▶ Business users in the Business role and WSRRUser role
- ▶ WSRR Advanced Lifecycle Edition administrators in WSRRAdmin role

## Rational Asset Manager

We used four communities: Shared services, Sports department, Clothing department, and Footwear department. Each community had a connection created to the WSRR system with the “Publish” type selected only.

We added Business Analysts and SOA Governance CoE members to all communities. All other users were placed in the community that matched their job roles.

Figure A-1 shows the permissions for the Sports department community.






















 bob	 Asset Consumer  Asset Producer  Forum admin
 david	 Asset Consumer  Asset Producer
 debra	 Asset Consumer  Asset Producer
 don	 Asset Consumer  Asset Producer
 larry	 Administrator
 lisa	 Asset Consumer  Asset Producer
 ramzan	 Asset Consumer  Asset Producer

Figure A-1 Permissions for the Sports department community

All users within the communities were able to produce and consume assets. The Asset Producer role was also modified to include the “Publishing administration” permission so that all users were able to publish documents from Rational Asset Manager to WSRR. Additionally, the business users were able to establish and manage the forums.

### Rational Asset Manager review process

All review processes used a review board to assess the overall result of the review and either approve or reject the asset. The review board also had the ability to tailor who was taking part in the review on an asset by asset basis.

Table A-1 shows the overall review process.

Table A-1 The review process for assets in Rational Asset Manager

Asset type	Review board members	Default additional reviewers
Business Solution	SOA Governance CoE	Business unit leader for LoB
Release	Business unit leader for LoB	Release manager for LoB
Specification	SOA Governance CoE	
Document of Understanding	Business unit leader for LoB	Release manager for LoB

The WSRR ALE Notification policy was attached to each review process created, for synchronization to WSRR.

### Theme customization

In a client setting, it is important that the user interfaces of the tools, WSRR and Rational Asset Manager, appear somewhat consistent. Therefore, within the scenarios, we customized the themes of WSRR and Rational Asset Manager slightly to offer consistency.

We modified the masthead of both tools as shown in Figure 15-18.



Figure 15-18 Masthead used in Rational Asset Manager and WSRR

## Charter validator

When writing the scenarios in this book, we discovered that the Charter document needed to be attached to the Charter relationship to satisfy the policies in WSRR, but it also needed to be attached to the Artifacts relationship so that the document link was replicated over to Rational Asset Manager.

Therefore, we developed a plug-in for WSRR that automatically correlates the Charter and Artifacts relationships on a Business Capability so that the Charter document did not need to be added as a target for both relationships.

We described how to load and activate the charter plug-in in “Charter plug-in” on page 365.

## Policy customization

In order for the business user to be able to modify the Business Capability to point to any new Capability Versions that they create, JKHLE modified the governance policies to allow the Business role to change approved Business Capabilities.



## Additional material

This book refers to additional material that you can download from the Internet as described next.

### Locating the Web material

The Web material associated with this book is available in softcopy on the Internet from the IBM Redbooks publications Web server. Point your Web browser at this Web site:

<ftp://www.redbooks.ibm.com/redbooks/SG247782>

Alternatively, you can go to the IBM Redbooks Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

Select **Additional materials**, and open the directory that corresponds with this IBM Redbooks form number, SG24-7782.

## Using the Web material

The additional Web material that accompanies this book includes the following files:

- ▶ **Export data/SG24-7782-00 WSRR Export [1.0].zip**  
WSRR prerequisites required to follow the scenarios.
- ▶ **Export data/SG24-7782-00 Rational Asset Manager Export [1.0].zip**  
Rational Asset Manager prerequisites required to follow the scenarios.
- ▶ **ValidatorPlugin**  
Folder containing the Charter plug-in used in the scenarios.
- ▶ **TopDownScenario**  
Folder containing documents referenced in Part 2, “Top-down scenario” on page 95.
- ▶ **BottomUpScenario**  
Folder containing documents referenced in Chapter 12, “Bottom-up scenario” on page 275.
- ▶ **Reporting**  
Folder containing reports for use in Chapter 15, “Reporting” on page 329.

## How to use the Web material

Create a subdirectory (folder) on your workstation, and unzip the contents of the Web material zip file into this folder.

### Scenario prerequisites

The JKHLE scenarios in this book expect that certain assets are already present in WSRR and Rational Asset Manager. To allow the user to replicate this situation, we have supplied a pair of import files (one file for WSRR and one file for Rational Asset Manager) with the book.

To import the prerequisites into WSRR:

1. Log in to WSRR as an administrator.
2. Select the **Import** option.
3. Select the SG24-7782-00 WSRR Export [1.0].zip file.

To import the prerequisites into Rational Asset Manager:

1. Log in to Rational Asset Manager as an administrator.



2. Select **Libraries** from the Administration page.
3. Choose to import the library by selecting the SG24-7782-00 Rational Asset Manager Export [1.0].zip file.
4. Enable and disassociate the library.

Note that the imported prerequisites will automatically be placed into the following three communities:

- ▶ Shared services
- ▶ Clothing
- ▶ Footwear

If the communities do not exist, or exist with a different name, these three communities will automatically be created and the assets will be placed in them.

## Referenced documents

The TopDownScenario and BottomUpScenario folders files that are provided contain the Web Services Description Language (WSDL) and XML Schema Definition Language (XSD) files that are referenced in the scenario sections to assist you when following the scenarios. Simply use the documents as required.

## Reporting

The Reporting folder contains the three reports that are created in Chapter 15, “Reporting” on page 329.

To use these reports:

1. Open your report tool.
2. Select the project to which the reports will be added.
3. Use the standard Eclipse import functionality to select the three reports.

## Charter plug-in

We have supplied the Charter plug-in and its source with this book. Add it as a standard validator in the WSRR configuration. Use this class name to specify in the validator properties configuration file:

```
com.ibm.redbook.a1e.CharterArtifactsCheck
```



# 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 368. Note that several of the documents referenced here might be available in softcopy only:

- ▶ *The Solution Designer's Guide to IBM On Demand Business Solutions*, SG24-6248
- ▶ *Patterns: SOA Foundation Service Creation Scenario*, SG24-7240
- ▶ *Implementing Technology to Support SOA Governance and Management*, SG24-7538
- ▶ *WebSphere Service Registry and Repository Handbook*, SG24-7386
- ▶ *Service Lifecycle Governance using WebSphere Service Registry and Repository V6.3*, SG24-7793
- ▶ *WebSphere Service Registry and Repository V6.3 Service Metadata Management and Governance*, REDP-4556
- ▶ *Integrating WebSphere Service Registry and Repository with WebSphere Process Server and WebSphere ESB*, REDP-4557
- ▶ *Integrating WebSphere Service Registry and Repository with WebSphere Message Broker and WebSphere MQ*, REDP-4558
- ▶ *Integrating WebSphere Service Registry and Repository with WebSphere DataPower*, REDP-4559
- ▶ *Integrating WebSphere Service Registry and Repository with Tivoli® Composite Application Manager*, REDP-4560
- ▶ *Integrating WebSphere Service Registry and Repository with Tivoli Security Policy Manager*, REDP-4561

## Online resources

These Web sites are also relevant as further information sources:

- ▶ IBM WebSphere Service Registry and Repository Version 6.3 Information Center  
<http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp>
- ▶ Introduction to SOA Governance  
<http://www.ibm.com/developerworks/library/ar-servgov/>
- ▶ SOA Governance and Service Lifecycle Management  
<http://www.ibm.com/soa/gov>
- ▶ WebSphere Service Registry and Repository Advanced Lifecycle Edition  
[http://www-01.ibm.com/software/integration/wsrr/ale/about/?S\\_CMP=rnav](http://www-01.ibm.com/software/integration/wsrr/ale/about/?S_CMP=rnav)
- ▶ IBM WebSphere Service Registry and Repository Version 6.3 Information Center  
<http://publib.boulder.ibm.com/infocenter/sr/v6r3/index.jsp>
- ▶ IBM Rational Asset Manager Version 7.1.1.1 Information Center  
<http://publib.boulder.ibm.com/infocenter/ramhelp/v7r1m1/index.jsp>
- ▶ *IBM Rational Software Rational Asset Manager v7.0 Capacity Planning and Configuration Guide*  
[http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/RAM\\_plan\\_config\\_guide\\_v4.0.5.pdf](http://download.boulder.ibm.com/ibmdl/pub/software/dw/rational/pdf/RAM_plan_config_guide_v4.0.5.pdf)
- ▶ Service-oriented modeling and architecture: How to identify, specify, and realize services for your SOA  
<http://www.ibm.com/developerworks/library/ws-soa-design1/>
- ▶ BIRT Project: Business Intelligence and Reporting Tools  
<http://www.eclipse.org/birt/phoenix>

## How to get IBM Redbooks publications

You can search for, view, or download IBM Redbooks and Redpaper publications, Technotes, draft publications, and Additional materials, as well as order hardcopy IBM Redbooks publications, at this Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

## Help from IBM

IBM Support and downloads

[ibm.com/support](https://ibm.com/support)

IBM Global Services

[ibm.com/services](https://ibm.com/services)

Archived





## Service Lifecycle Governance with IBM WSRR Advanced Lifecycle Edition

(0.5" spine)  
0.475" <-> 0.875"  
250 <-> 459 pages









# Service Lifecycle Governance with IBM WebSphere Service Registry and Repository Advanced Lifecycle Edition

**Implement service  
governance**

**Use WSRR Advanced  
Lifecycle Edition  
effectively**

**Understand  
lifecycyle  
management**

The purpose of service-oriented architecture (SOA) governance is to act as the catalyst between business governance and IT governance by ensuring the most optimal business value for services investments. Within SOA governance, there is service governance, which ensures that organizations build the right services, in the right way, at the right time, and then manage and reuse those services effectively. Service governance oversees the processes of proactively identifying, assessing, building, and managing high-value business services, those services that provide the greatest return on investment. Service governance requires a federated, end-to-end approach to service lifecycle governance and management, including knowing what needs to be strictly governed and where a lighter touch is more appropriate.

As SOA evolves to be a fundamental and critical enabler for other efforts, such as IBM smarter planet and cloud computing initiatives, it becomes imperative to ensure quality SOA and service governance capabilities. These capabilities drive discipline and value for other areas, such as the governance of provisioning of assets, impact analysis, and asset/service portfolio management.

This book identifies the key functions and capabilities that are required for service governance based on field best practices and client scenarios. This information gives IBM the ability to deliver higher business value to our clients in more consumable and fit for purpose delivery methods that match the way our clients operate.

**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](http://ibm.com/redbooks)