Govern service assets at development time using IBM Rational software
White paper
August 2007

Rational® software

SOA governance with IBM Rational Asset Manager software.

Steve Weaver, architecture management marketing engineer, IBM Rational

Contents

Introduction

The success of service-oriented architecture (SOA) in terms of software development and delivery is dependent on a strong governance environment supporting the effort to develop and deploy services and service-based applications and frameworks.

Developing an end-to-end governance approach involves creating a decision-making structure from executives such as the chief information officer (CIO) down to the individual business areas and IT organizations. In addition, SOA governance controls the end-to-end service lifecycle both during the development process and at run time.

SOA governance is about setting up the policies, procedures and processes that are needed to allow for efficient and effective decision making throughout a business and IT organization. Once the governance process has been established, it is executed through the management of the SOA service lifecycle.

This paper discusses some of the challenges around SOA governance in the context of a lifecycle of services, and introduces IBM Rational® Asset Manager software as a solution to those challenges, illustrating the principles of IBM's development-time metadata management strategy in the context of SOA governance.

Defining SOA governance

To better understand how SOA governance fits within the context of overall corporate governance, it is first necessary to define what we mean by governance in general, and by IT governance and SOA governance in particular.

Governance

There are two fundamental characteristics of governance. The first characteristic involves the processes established by a business to determine who is empowered to make certain decisions. The second characteristic includes the mechanisms and policies that are used by the business to measure and control the way those decisions are implemented. The first characteristic can be thought of as the process by which the development and delivery of services are to be managed, and the second can be thought of as the implementation of that process using tools.

Governance is usually viewed as an executive function, while the execution of the governance framework is the responsibility of project managers and, ultimately, the analysts, architects, developers and testers who implement the policies. In this paper, we will discuss the governance of SOAs, which can be thought of as an extension of IT governance. First, let's define both IT and SOA governance.

IT governance

IT governance aligns IT activities with the goals of the business as a whole. This subset of governance includes the decision-making rights associated with IT investment, as well as the policies, practices and processes used to measure and control the way IT decisions are prioritized and executed. IBM Rational software provides key tools to help automate the process of IT governance.

SOA governance

Adoption of an SOA raises new issues both in IT decision rights and in measurement and control. IT organizations must consider which operation is best suited to deploy different services under varying conditions, and they must be able to monitor that the services are in fact enhancing business value. SOA governance is an extension of IT governance that focuses on the lifecycle of services and composite applications in an organization's SOA. Deploying an SOA frequently serves as a catalyst for an organization to start thinking about improved corporate and IT governance, and how to best implement SOA governance practices. In this paper, we will discuss the implementation of service lifecycle management through Rational Asset Manager software.

SOA governance and IBM

In March 2006, IBM announced a view of SOA governance that included extensions of IT governance focused on the lifecycle of services to help ensure the business value of SOA, establish decision rights to empower people and policies, and implement measurement and control mechanisms to enable people.

Many people, processes and tools can be part of an overall end-to-end SOA governance environment. Governance processes are developed using the IBM SOA Governance and Management Method with the Rational Method Composer plug-in; then those policies and procedures are automated using tools such as IBM Rational Portfolio Manager, Rational Asset Manager and IBM WebSphere® Service Registry and Repository software.

To understand the SOA governance environment, there are two parts that need to be considered. The first is the SOA governance and service lifecycle management processes (figure 1). The second is the execution of the processes and the management and monitoring of the results. In this case, there are two types of results: the services that come out of the development process, and the decisions that are made related to the services. The SOA governance processes are monitored via the SOA governance framework, while the released services are managed via the manage component of the SOA lifecycle.

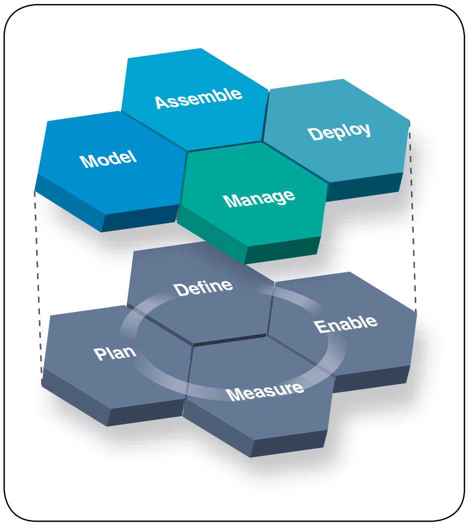


Figure 1: The SOA governance process (bottom) and the service lifecycle management process (top)

SOA governance process framework

Let's address the SOA governance process framework first. Plan, define, enable and measure are all about establishing the end-to-end SOA governance process. This will involve vertical management decision-making process as well as horizontal SOA lifecycle management processes. The SOA lifecycle then has the SOA governance framework as the driver for the overall processes, which affects how decisions are made and how policies for the SOA lifecycle are developed in the SOA governance framework. Guidance for the SOA governance framework is provided in the IBM Rational Unified Process® (IBM RUP®) Service-Oriented Modeling Architecture (SOMA) plug-in to the Rational Method Composer tool.

Service lifecycle management process

The service lifecycle management process may be only one of the processes governed by the SOA governance framework, but it is the key process for the development and management of services. Services are created within the model and assemble stages of the SOA lifecycle, and are managed by Rational Asset Manager software before they are deployed into a production environment (the deploy stage of the SOA lifecycle), where they can be managed by WebSphere Service Registry and Repository software. Once deployed, services are controlled in the manage stage of the SOA lifecycle. The overall process is a closed loop, since services can be updated during their lifetime, so they would end up reentering the process through the model phase.

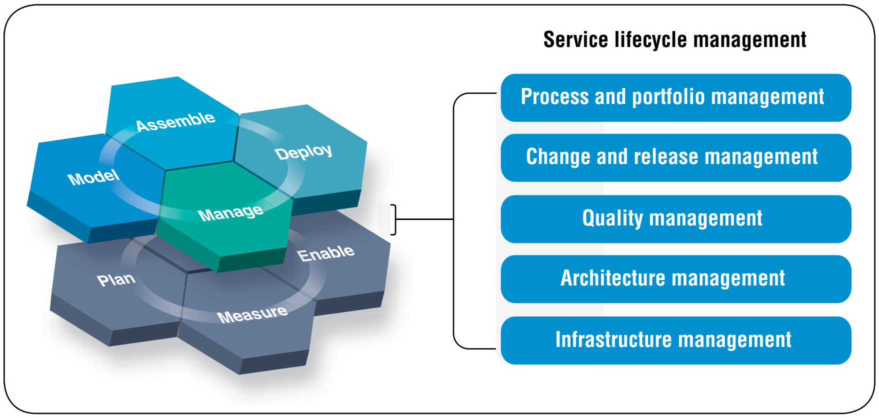

Service lifecycle management
Assemble
Deploy
Model
Manage
Enable
Plan
Measure
Process and portfolio management
Change and release management
Quality management
Architecture management
Infrastructure management

Figure 2: Service lifecycle management components and Rational capabilities

IBM Rational offers solutions that help with service lifecycle management (figure 2), process and portfolio management (Rational Portfolio Manager, IBM Rational RequisitePro® and Rational Method Composer software), change and release management (IBM Rational ClearCase® and IBM Rational ClearQuest® software), quality management (IBM Rational Performance Tester, IBM Rational Functional Tester and IBM Rational SOA Tester software), architecture management (IBM Rational Software Architect, IBM Rational Application Developer software) and now infrastructure management (Rational Asset Manager software). The focus of this paper is on infrastructure management and the Rational Asset Manager solution.

Overview of IBM Rational Asset Manager software

IBM offers a number of metadata repositories designed to help solve the challenges of ensuring adequate governance of the service lifecycle. Each one of the repositories is targeted at a specific user community, supporting the tasks and processes important to that community, and offering the content relevant to those tasks and that community. WebSphere Service Registry and Repository is a repository for managing SOA metadata relevant to deployed services. IBM Tivoli® Change and Configuration Management Database is a repository for managing system and service management metadata, while a metadata repository for addressing the needs of an information management audience is IBM WebSphere Metadata Server software. IBM believes that is it important to have different repositories addressing different communities, and furthermore, IBM acknowledges the need for federation of information between those repositories—making common content available in the appropriate form to the different communities.

Featuring a service asset repository to help govern the service lifecycle, Rational Asset Manager software is a key enabler of SOA governance. The seamless search and publish capabilities across all phases of the SOA lifecycle foster reuse of services and enrich connectivity with dynamic and efficient interactions between deployed services (through WebSphere Service Registry and Repository software) and services in development. The robust asset management functions for the entire software lifecycle make Rational Asset Manager software an essential tool for the success of SOA. Regardless of the level of SOA maturity in the enterprise, Rational Asset Manager software can potentially bring immediate benefits to accelerate an organization's adoption of SOA.

Managing reusable assets for development

An asset is defined as a collection of artifacts that provide a solution to a problem for a given context. Typically, an asset may have variability points that allow users to customize the asset by setting various parameters. An asset that can be treated in this way is called a template. Assets also include instructions or rules for their use in order to minimize the time developers need to discover, analyze, consume and test the asset. Assets also describe the development and business contexts in which the asset can and should be reused. Today, Rational Asset Manager software is implemented with this definition in mind.

Rational Asset Manager is a development-time asset repository that helps manage assets relevant to development roles such as technical managers, analysts, architects, developers and testers. The Rational Asset Manager tool governs the assets as they are submitted; categorizes the assets; provides access control to the assets; and measures the activity level of assets in terms of their usage.

An asset has several major characteristics:

- *It's a collection of one or more artifacts (e.g., files, binaries, models, tests).*
- *It has relationships to other assets (e.g., dependencies, aggregation).*
- *It can be classified (i.e., tagged with values and terms).*
- *It has usage measurements (e.g., who is using the asset, what defects it has).*
- *It controls user access (i.e., authorization must be given to perform certain changes).*
- *It works under policies (e.g., descriptions of proper structures and content).*

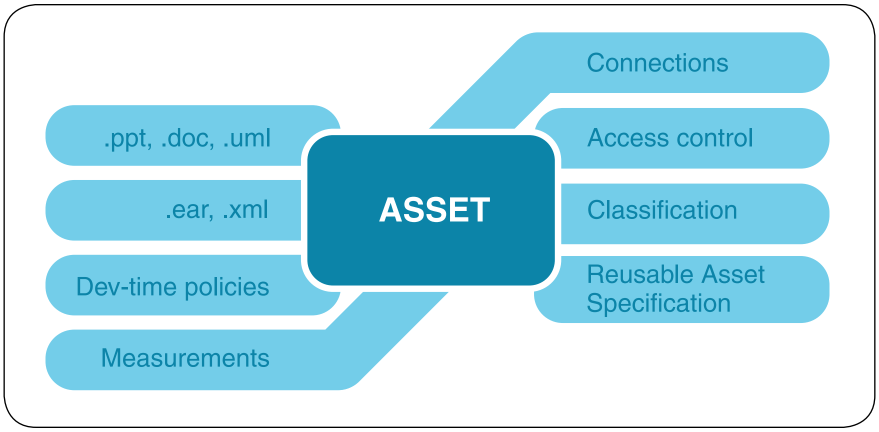


Figure 3: An asset may contain a number of artifacts to provide a solution to a problem for a given context.

Examples of assets include applications, components or more fine-grained elements such as business process models, patterns or a Microsoft® Word design document template.

This extensible structure for assets permits enterprises to specify the nature of and control over their high-value assets. Enterprises can specify architectural structures and control through extensible asset types and other mechanisms, which are then managed and enforced by Rational Asset Manager software.

Rational Asset Manager manages metadata on assets to support asset management scenarios. Metadata can be defined as data about data, or more specifically, information (data) about a particular piece of content (data). An item of metadata may describe an individual content item or a collection of content items. Metadata is used to facilitate the understanding, use and management of data. The metadata required for this will vary with the type of data and the context of use. For example, in the context of a library, where the data is the content of the titles stocked, metadata about a title might include a description of the content, the author, the publication date and the physical location of the work. The metadata stored in Rational Asset Manager includes the obvious items such as the asset's name, description, version and state. Other metadata may include the name, description, version and reference (or location) of the contained artifacts.

To support these scenarios, Rational Asset Manager software supports the Reusable Asset Specification (RAS) from the Object Management Group (OMG) as the core structure for asset metadata.

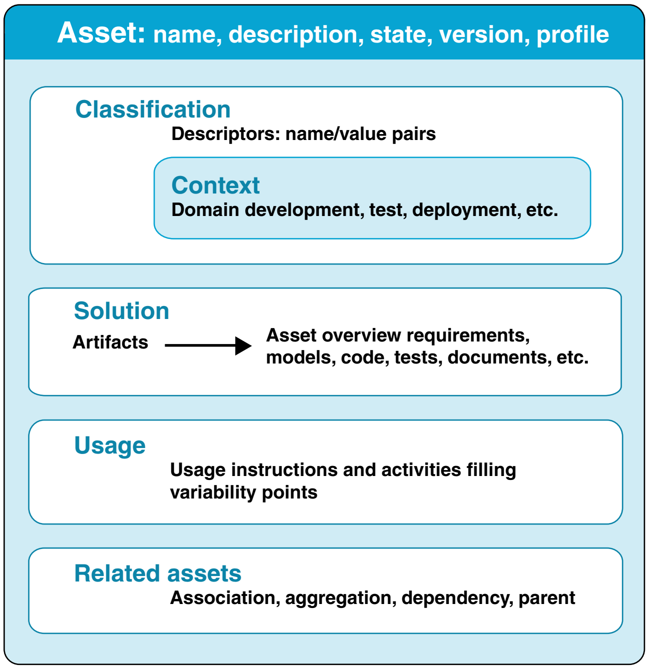

Asset: name, description, state, version, profile
Classification
Descriptors: name/value pairs
Context
Domain development, test, deployment, etc.
Solution
Artifacts
Asset overview requirements, models, code, tests, documents, etc.
Usage
Usage instructions and activities filling variability points
Related assets
Association, aggregation, dependency, parent

Figure 4: The RAS metadata structure

As figure 4 shows, a RAS asset has a classification section that facilitates searching and browsing. This section might include simple name/value pair descriptors and a declaration of contexts, such as a specific domain, development or deployment context.

The solution section shown in figure 4 is the meat of the asset; it describes the collection of artifacts that provides a solution.

The usage section provides guidance on applying and customizing the asset using variability points. Some usage information may be automated using scripts and wizards, which are stored in the solution section with the other asset artifacts.

The related assets section defines the asset's relationships to other assets and helps create collections, or families, of assets to form larger-grained solutions.

Using RAS in the Rational Asset Manager solution, the enterprise can configure multiple asset types to meet their particular needs. In the case of SOA, an example of an asset could be a service. Let's look a little more deeply at how service assets may be structured.

A service can be defined as a single asset type containing all necessary artifacts, or it can be deconstructed into many asset types such as service interface, service implementation and service test. When defining asset types, one should consider who will be using the asset. This means artifacts and relationships must also be defined (figure 5). Assets are created with a specific type, and they possess the characteristics defined by the asset type.

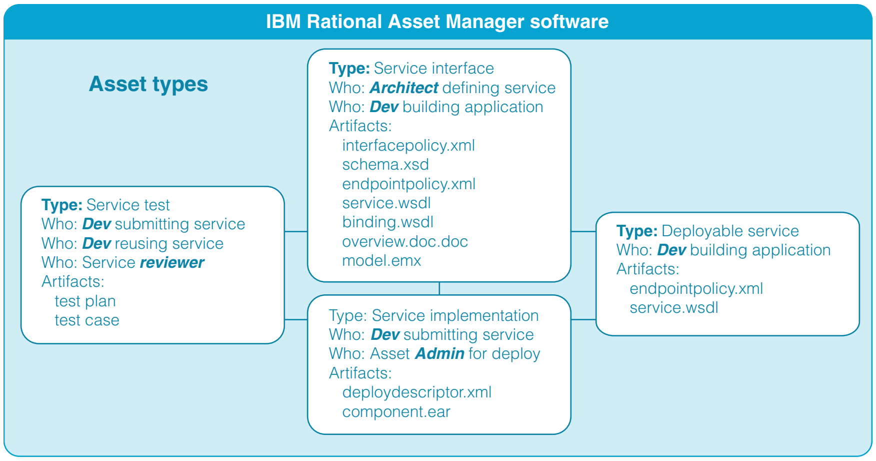


Figure 5: An example of related assets defining a service in Rational Asset Manager software

Rational Asset Manager software and SOA governance

Rational Asset Manager software is designed to effectively govern the development of all assets, including service assets. In this section, an example scenario illustrates the process.

Building an effective SOA governance framework

SOA governance, in addition to the more traditional software development and delivery lifecycle checkpoints and role-based review sign-offs, focuses on the creation, communication and enforcement of service policies. Service policies consist of a set of constraints and capabilities that govern how services and their consumers interact.

SOA governance requires that organizations take business policies, typically in written form, and transform them into metadata-based rules that can help automate the process of validating and enforcing compliance with those policies in both design-time and run-time environments.

Companies must then manage policies through their entire lifecycle. In general, policy lifecycle management within SOA focuses on ensuring the quality, performance and applicability of available services, enabling service consumers to discover and reuse services as well as other artifacts; manage service versions; handle the security of services and other SOA artifacts; and assess and manage the impact of change across all service consumers. Managing policies also includes providing visibility into whether people are following policies, as well as handling policy infractions. Such policy management tasks are also an inherent aspect of IT governance.

Governance maximizes the value of SOA in aligning business needs with IT direction. Activities needed to establish, maintain and enhance an effective SOA governance framework are depicted as a lifecycle consisting of four phases: plan, define, enable and measure.

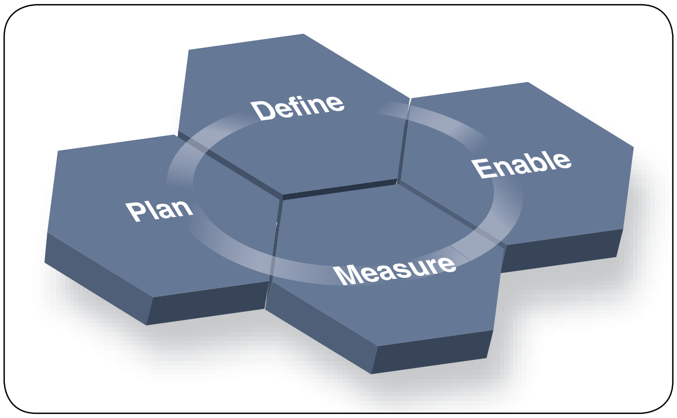


Figure 6: The SOA governance framework

Plan

The planning phase in an SOA governance framework focuses on understanding the overall scope of the governance opportunity within the organization, and identifying areas for improvement.

Define

Once the opportunities for improved governance are identified, business and IT people can work together to define and modify the current governance policies and mechanisms.

Enable

Solutions to governance needs are put into action during this phase. These solutions may include deployment of new or enhanced governance policies. It is likely that communication and education mechanisms will be rolled out to entrench the new policies within the business and the IT communities. Governance activities within this phase influence how an SOA is deployed by enabling the policy enforcement infrastructure.

Measure

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

Applying SOA governance

SOA governance can enable organizations to maximize the business benefits of SOA, which can include increased process flexibility, improved responsiveness and reduced IT maintenance costs. It also mitigates many of the business risks inherent in SOA adoption by establishing decision rights, guiding the definition of appropriate services, managing assets and measuring effectiveness.

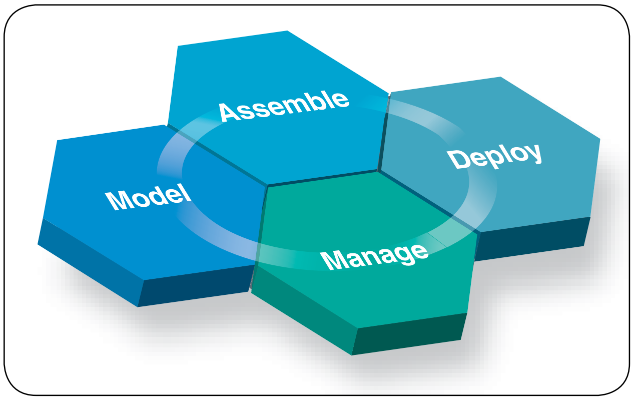
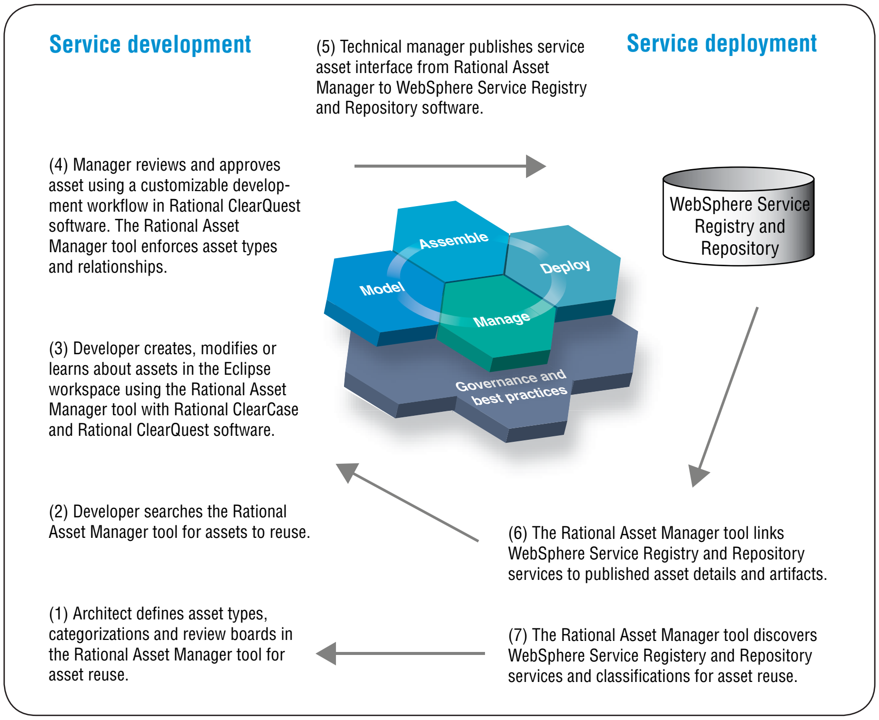


Figure 7: The service lifecycle

The following scenario demonstrates how Rational Asset Manager software addresses common SOA governance challenges.



Service development

Service deployment

(1) Architect defines asset types, categorizations and review boards in the Rational Asset Manager tool for asset reuse.

(2) Developer searches the Rational Asset Manager tool for assets to reuse.

(3) Developer creates, modifies or learns about assets in the Eclipse workspace using the Rational Asset Manager tool with Rational ClearCase and Rational ClearQuest software.

(4) Manager reviews and approves asset using a customizable development workflow in Rational ClearQuest software. The Rational Asset Manager tool enforces asset types and relationships.

(5) Technical manager publishes service asset interface from Rational Asset Manager to WebSphere Service Registry and Repository software.

(6) The Rational Asset Manager tool links WebSphere Service Registry and Repository services to published asset details and artifacts.

(7) The Rational Asset Manager tool discovers WebSphere Service Registery and Repository services and classifications for asset reuse.

WebSphere Service Registry and Repository
Assemble
Model
Deploy
Manage
Governance and best practices

Figure 8: Rational Asset Manager service development scenario

Manage: establishing decision rights

An SOA initiative can be derailed if there is no governance framework in place to clearly identify roles, responsibilities and decision rights. Rational Asset Manager software can provide comprehensive capabilities around defining roles and responsibilities.

Rational Asset Manager software helps administrators create communities that define assets used by specific projects. Users and groups are then created, providing secure access to a community's assets. Users can also be assigned roles, which are an important step in defining who is allowed to perform activities like browsing, creating, updating, submitting or reviewing assets.

In addition, the community administrator can set up review processes per asset type and category. Review processes can be customized either through the out-of-the-box review process or with Rational ClearQuest software. A typical review process involves the following elements:

- *Support for review boards and individual reviewers*
- *Support for as-is states*
- *Automatic notifications to reviewers*

Figure 9 represents an example of the asset review process, which is defined through the Web GUI, as shown in figure 10.

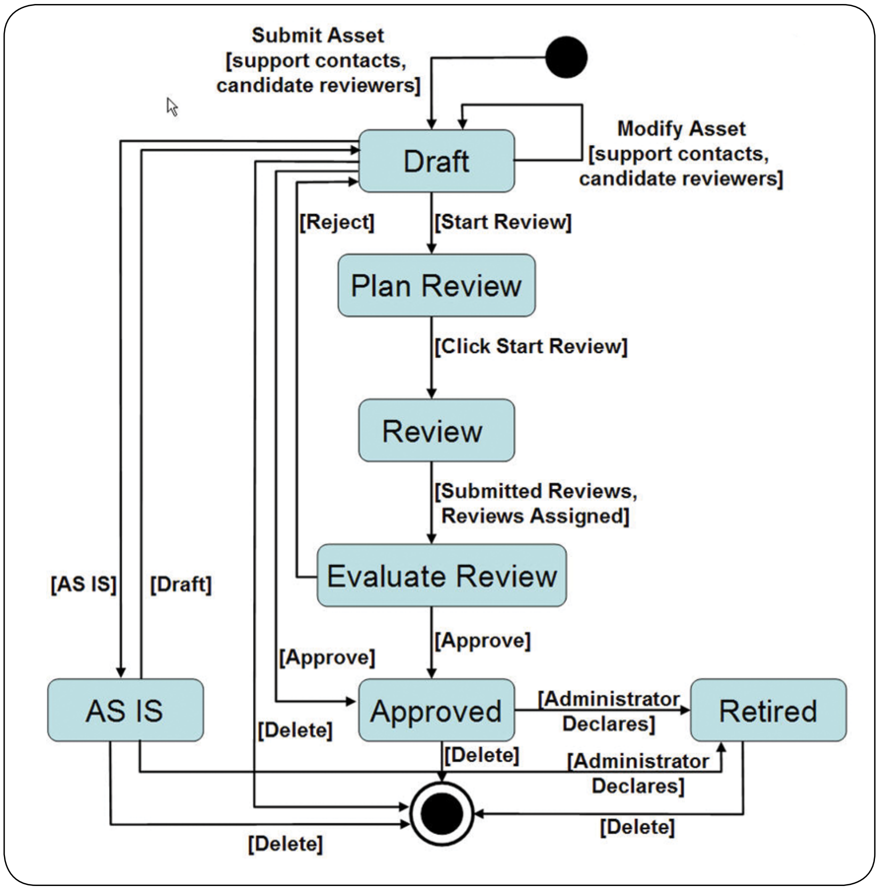
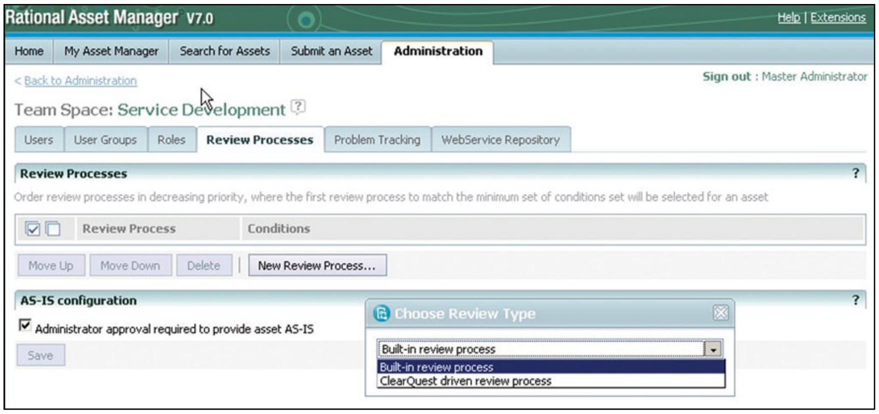


Figure 9: An example asset review process



Order review processes in decreasing priority, where the first review process to match the minimum set of conditions set will be selected for an asset

☑ Administrator approval required to provide asset AS-IS

Choose Review Type: Built-in review process

Figure 10: Rational Asset Manager Web GUI for defining the asset review process

Model: defining appropriate service assets

Governance issues can impact the approach to development. Suppose an insurance company decides to expand into new geographies that have different regulatory requirements. An SOA is critical to successfully implementing this strategy because it can allow shared reusable business services to be bundled or unbundled as required to help enable compliance in each geographical area. Effective SOA governance can help answer the following types of questions:

- *What are the potential applications (service consumers) that will reuse this service?*
- *What are the common business services that are needed?*
- *What services already exist and are candidates for reuse?*
- *Which services can be shared, and under what rules and circumstances can they be shared?*
- *Which polices and standards can be factored as common across the geographical regulatory jurisdictions? Which are separate?*
- *Can the differences be isolated to maximize consistency while still being compliant?*

With SOA governance, the insurance company can potentially establish—and communicate—the policies that need to be enforced. These decisions can then be consciously built into the development process used to create applications and services. Because an SOA allows policies to be isolated from the code, the insurance company is able to tailor the services by geography, yet still lower overall development costs by maximizing component reuse.

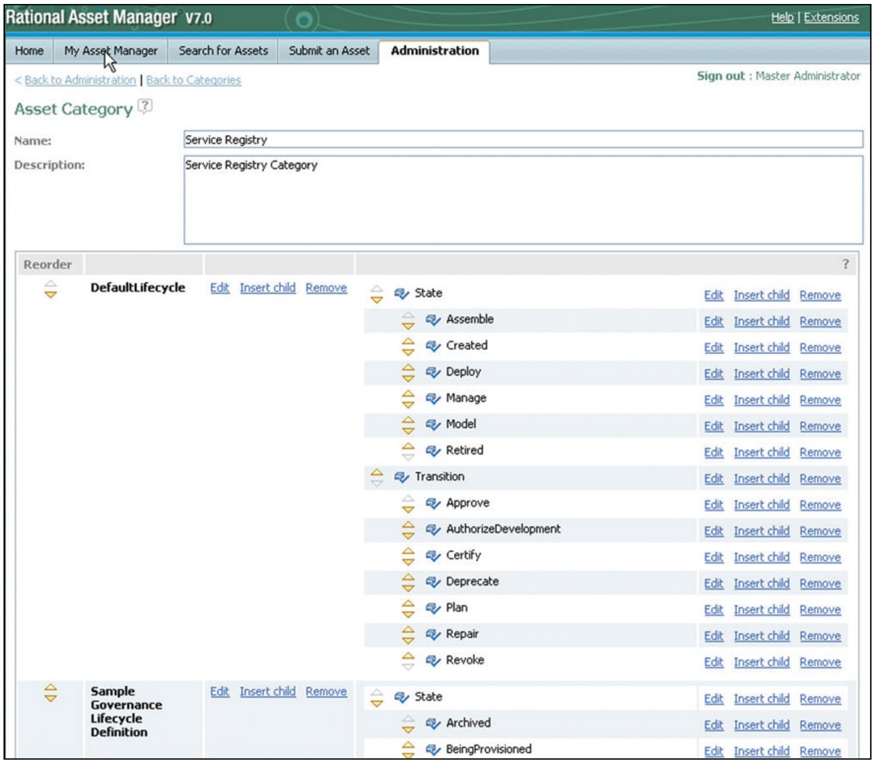
Rational Asset Manager software can allow for the definition of governance policies through a combination of asset categorization and review. We've discussed the asset review process above, but it should be noted that each asset type can have its own review process defined to answer some of the questions posed earlier.

In addition, asset types can be created to handle any asset. An asset type can be almost anything—an application, component, pattern or service—and specifies:

- *Type name and description*
- *Required artifacts*
- *Required relationships*
- *Attributes (optional or required)*
- *Which categories are allowed*

Validation of asset type constraints is run on asset submission to help ensure that standards for assets are enforced.

Asset categorization is another way assets can be governed. Categories for asset classification are configured via the Rational Asset Manager Web GUI (shown in figure 11) or an XML file. Assets can belong to any number of categories, and categories can be used for faceted search, browsing and access control.


Rational Asset Manager V7.0
Help | Extensions
Home
My Asset Manager
Search for Assets
Submit an Asset
Administration
< Back to Administration | Back to Categories
Sign out : Master Administrator

Asset Category

Name: Service Registry

Description: Service Registry Category

Reorder				?
	DefaultLifecycle	Edit Insert child Remove	State	Edit Insert child Remove
			Assemble	Edit Insert child Remove
			Created	Edit Insert child Remove
			Deploy	Edit Insert child Remove
			Manage	Edit Insert child Remove
			Model	Edit Insert child Remove
			Retired	Edit Insert child Remove
			Transition	Edit Insert child Remove
			Approve	Edit Insert child Remove
			AuthorizeDevelopment	Edit Insert child Remove
			Certify	Edit Insert child Remove
			Deprecate	Edit Insert child Remove
			Plan	Edit Insert child Remove
			Repair	Edit Insert child Remove
			Revoke	Edit Insert child Remove
	Sample Governance Lifecycle Definition	Edit Insert child Remove	State	Edit Insert child Remove
			Archived	Edit Insert child Remove
			BeingProvisioned	Edit Insert child Remove

Figure 11: Asset categorization in Rational Asset Manager software

For example, a taxonomy for our insurance company example might be as follows:

- *Financial reporting*
  - *Stockholders equity*
  - *Financial position*
  - *Income statement*

- *Health insurance*
  - *Public*
    - *Claims processing*
    - *Tax*
    - *Social security*
  - *Private*
    - *Mandatory*
    - *Employment group*
    - *Community rated*
    - *Risk rated*

Assemble: managing service assets

SOA governance issues also arise when dealing with existing systems. For example, faced with increasing competition, and in an effort to increase customer satisfaction, a large retailer institutes a new billing policy that charges customers when an order is shipped. The system replaces the old policy that charged customers when an order was placed. The development team finds it hard to determine which applications and other services are impacted by this change. But facing an implementation deadline, it forges ahead, making changes to those applications it knows are affected, and putting the new billing policy into production. Unfortunately, another line-of-business application was using the previous versions of the services that make up the billing policy. This causes unpredictable outages, customer complaints and revenue losses. Although each line of business adopted SOA, each also implemented its own unique billing policy solutions, making it difficult to deploy a company-wide billing policy.

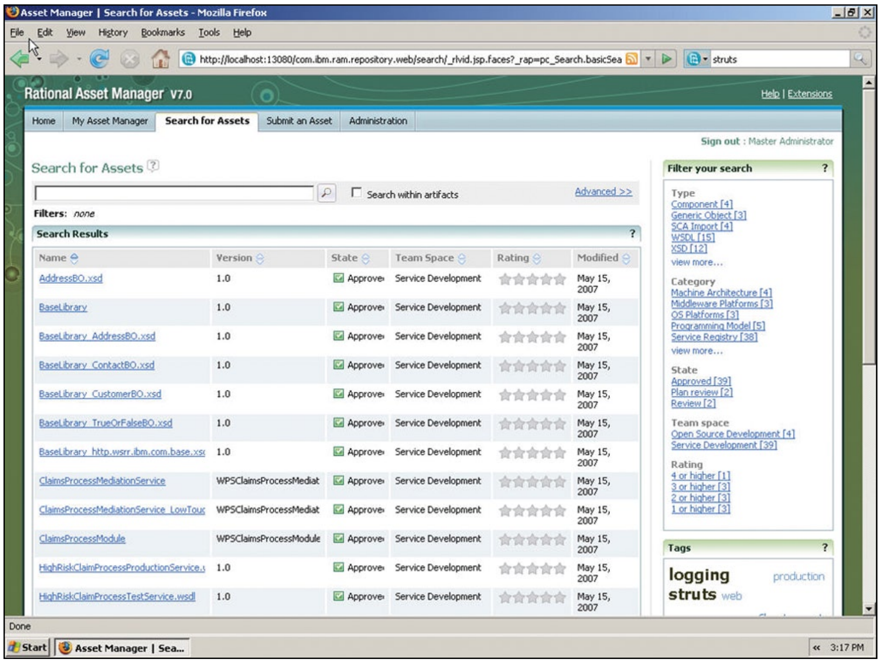
The organization was successful in setting up the governance arrangements to enable service reuse, but it failed to institute a process by which services are versioned and updated within the shared infrastructure. It also failed to establish mechanisms for monitoring who used the services. Such ineffective asset management can lead to service proliferation and derailment of the reuse strategy. And without the ability to assess the impact of changing a service asset, unanticipated breakage can negatively impact the business.

Key governance issues that must be addressed in this kind of scenario include:

- *How can the shared business services and other enterprise assets be organized so they can be effectively reused later?*
- *Who is allowed to change a service reused by others?*
- *Who is using a given service?*
- *Who needs to approve the change?*
- *Who will be responsible for funding changes if the infrastructure needs to be upgraded to meet specific performance or availability requirements later?*

SOA governance can help the retailer ensure that services are effectively reused, updated and tracked within the shared infrastructure. It can also help the retailer understand the impact of change across the organization from both a business and a technical point of view. Moreover, it can help the retailer better plan for and implement those changes.

Rational Asset Manager software is designed to automate these scenarios through extensive search and evaluation capabilities. A developer can search using categories, facets or keywords, and the search can include full access to services indexed by WebSphere Service Registry and Repository software. Searches are indexed for fast retrieval and can be performed through either a Web or Eclipse client.



http://localhost:13080/com.ibm.ram.repository.web/search/_rlvid.jsp.faces?_rap=pc_Search.basicSea

struts

Sign out : Master Administrator

Search for Assets

□ Search within artifacts

Advanced >>

Filters: *none*

Search Results

Name	Version	State	Team Space	Rating	Modified
AddressBO.xsd	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
BaseLibrary	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
BaseLibrary_AddressBO.xsd	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
BaseLibrary_ContactBO.xsd	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
BaseLibrary_CustomerBO.xsd	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
BaseLibrary_TrueOrFalseBO.xsd	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
BaseLibrary_http.wsrr.ibm.com.base.xs	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
ClaimsProcessMediationService	WPSClaimsProcessMediat	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
ClaimsProcessMediationService_LowTouc	WPSClaimsProcessMediat	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
ClaimsProcessModule	WPSClaimsProcessModule	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
HighRiskClaimProcessProductionService.	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007
HighRiskClaimProcessTestService.wsdl	1.0	☑ Approve	Service Development	☆☆☆☆☆	May 15, 2007

Filter your search

Type
Component [4]
Generic Object [3]
SCA Import [4]
WSDL [15]
XSD [12]
view more...

Category
Machine Architecture [4]
Middleware Platforms [3]
OS Platforms [3]
Programming Model [5]
Service Registry [38]
view more...

State
Approved [39]
Plan review [2]
Review [2]

Team space
Open Source Development [4]
Service Development [39]

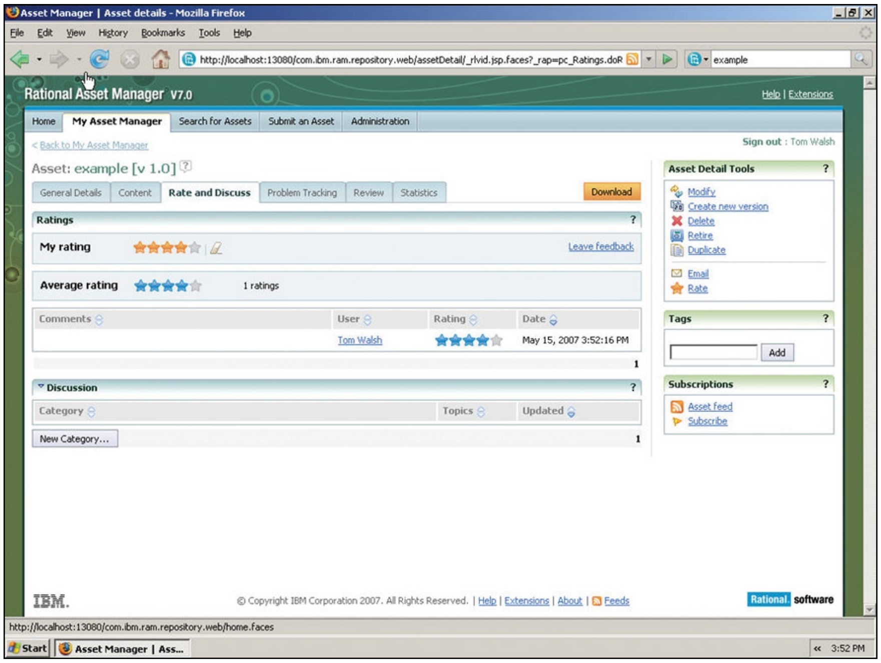
Rating
4 or higher [1]
3 or higher [3]
2 or higher [3]
1 or higher [3]

Tags

logging production
struts web

Figure 12: Searching for assets

More important, the developer can evaluate the service asset before downloading and using it. The developer can browse the asset to the artifact level, and the ability to read ratings, feedback and statistics can help in asset evaluation.



Comments	User	Rating	Date
	Tom Walsh	★★★★☆	May 15, 2007 3:52:16 PM

Category	Topics	Updated

Figure 13: Rational Asset Manager feedback for asset ratings

Assemble: creating or updating service assets

Once the developer has evaluated the asset, has and is assuming the correct permissions, the asset can be downloaded in one click through either a Web or Eclipse client. This action inserts the asset directly into an Eclipse-based integrated development environment (IDE) such as Rational Application Developer, Rational Software Architect or IBM WebSphere Integration Developer software.

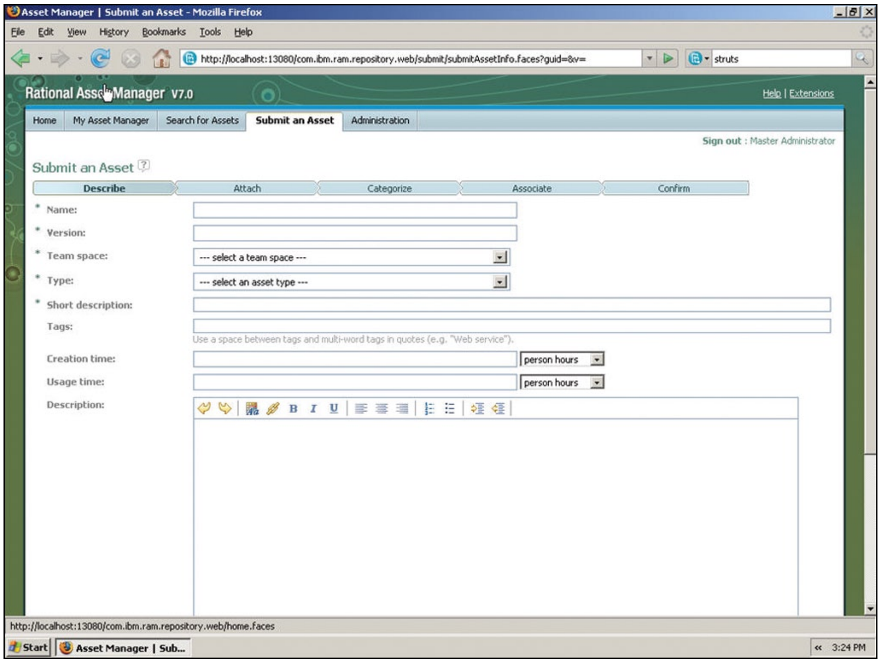
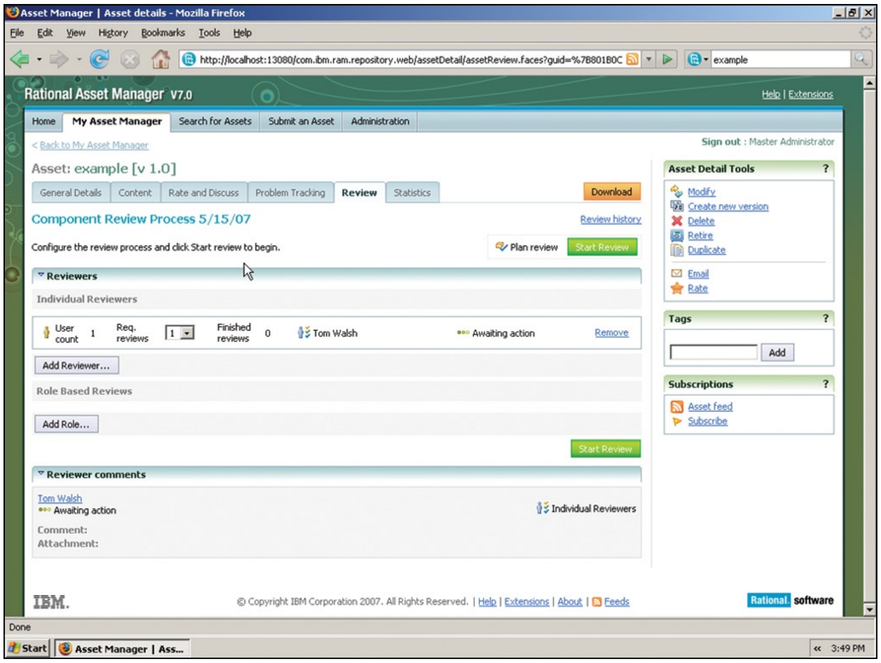

Asset Manager | Submit an Asset - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:13080/com.ibm.ram.repository.web/submit/submitAssetInfo.faces?guid=&v=
struts
Rational Asset Manager V7.0
Help | Extensions
Home
My Asset Manager
Search for Assets
Submit an Asset
Administration
Sign out : Master Administrator
Submit an Asset
Describe
Attach
Categorize
Associate
Confirm
* Name:
* Version:
* Team space:
--- select a team space ---
* Type:
--- select an asset type ---
* Short description:
Tags:
Use a space between tags and multi-word tags in quotes (e.g. "Web service").
Creation time:
person hours
Usage time:
person hours
Description:
http://localhost:13080/com.ibm.ram.repository.web/home.faces
Start
Asset Manager | Sub...
3:24 PM

Figure 14: Submitting an asset

If a developer is creating a new asset, that process is automated through a wizard in both the Web and Eclipse user interfaces. The developer can choose asset type, categories, relationships and artifacts to upload, and can specify additional tags for searching. The wizard guides the user based on how the asset type is defined.

Assemble: review asset submission

Assuming our developer has submitted an asset, it will automatically be put into the review process defined earlier. During the review, the asset manager or review board determines and notifies reviewers. Reviewers evaluate the asset and submit responses. The asset manager or review board then approves or rejects the asset for publishing based on submitted and evaluated reviews. Finally, the system notifies the relevant parties.



Configure the review process and click Start review to begin.

User count 1 Req. reviews 1 Finished reviews 0 Tom Walsh Awaiting action Remove

Tom Walsh
Awaiting action
Individual Reviewers
Comment:
Attachment:

Figure 15: Asset review via a Web GUI

The asset review process addresses issues organizations face in implementing a governance policy for service assets. It:

- *Validates and verifies assets for a specified context and scope of reuse.*
- *Helps ensure that assets have met enterprise rules, policies and governance.*
- *Removes the lack of awareness of where the asset is in the workflow.*

Deploy: integration with service asset run-time repositories

The Rational Asset Manager tool synchronizes with WebSphere Service Registry and Repository software to provide coordination of asset and service metadata. To enable this synchronization, the administrator configures one or more WebSphere Service Registry and Repository connections for a Rational Asset Manager community.

For example, the administrator may configure one connection to a WebSphere Service Registry and Repository test registry, and another connection to a WebSphere Service Registry and Repository production registry, as shown in figure 16. Each connection needs information such as the address and port of the host, the WebSphere Service Registry and Repository login credentials, and the owner of the assets. Each asset in Rational Asset Manager software has an owner, so when WebSphere Service Registry and Repository shadow assets are created, the proper asset owner will be set.

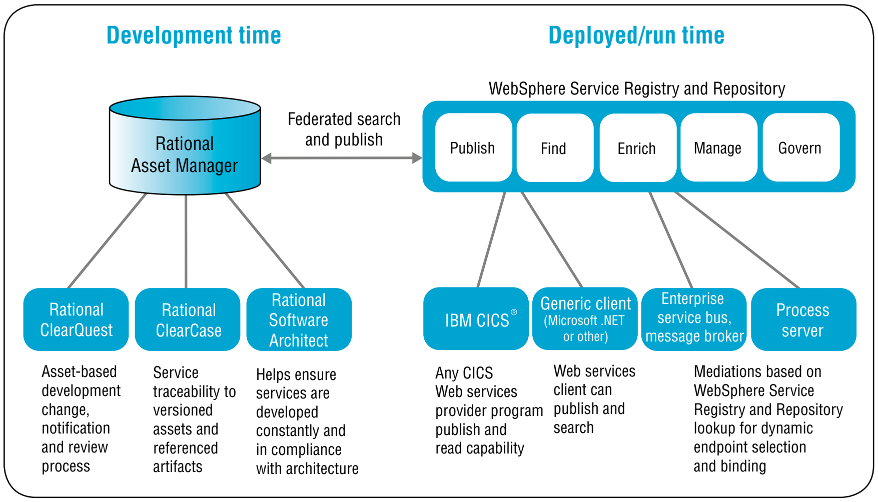

Development time
Deployed/run time
WebSphere Service Registry and Repository
Rational Asset Manager
Federated search and publish
Publish
Find
Enrich
Manage
Govern
Rational ClearQuest
Rational ClearCase
Rational Software Architect
IBM CICS®
Generic client (Microsoft .NET or other)
Enterprise service bus, message broker
Process server
Asset-based development change, notification and review process
Service traceability to versioned assets and referenced artifacts
Helps ensure services are developed constantly and in compliance with architecture
Any CICS Web services provider program publish and read capability
Web services client can publish and search
Mediations based on WebSphere Service Registry and Repository lookup for dynamic endpoint selection and binding

Figure 16: Rational Asset Manager and WebSphere Service Registry and Repository federation

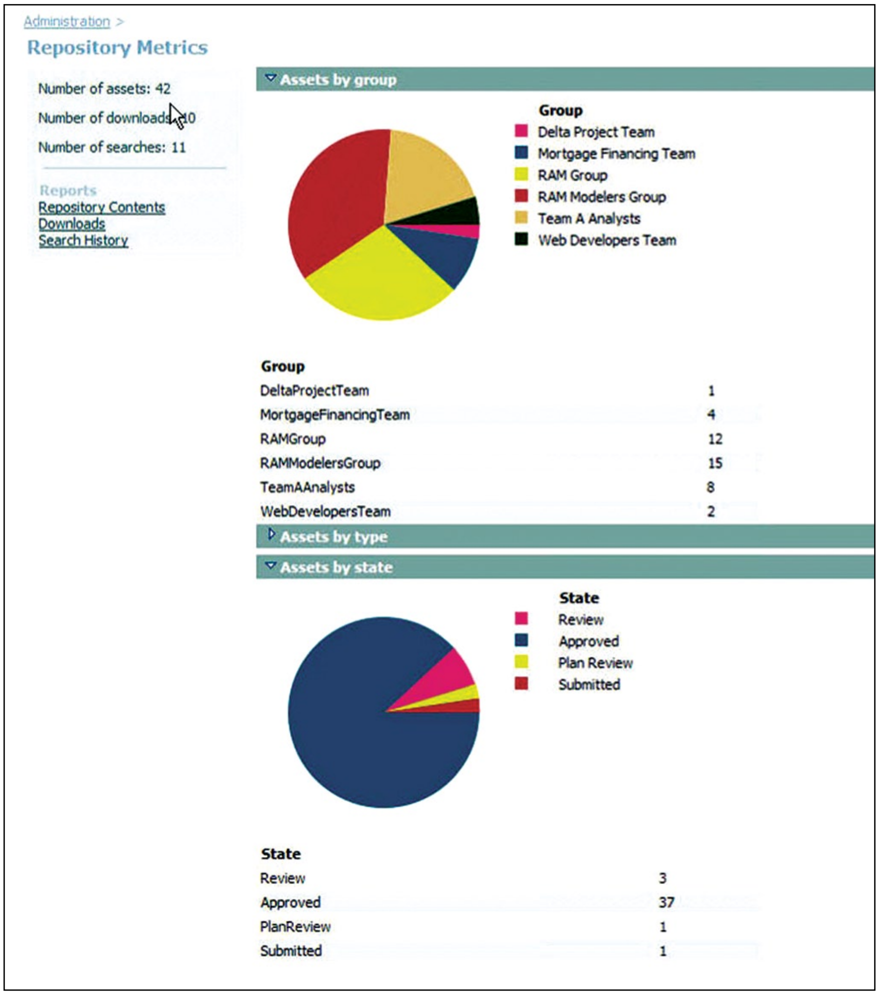
Rational Asset Manager software is designed to allow service asset artifacts to be published to WebSphere Service Registry and Repository software when the community is configured with one or more WebSphere Service Registry and Repository connections and there is at least one Web Service Description Language (WSDL) file, XML Schema Definition (XSD) file or XML file in a service asset. When these conditions are met, the *Publish to service registry* option becomes visible in the service asset details. If more than one WebSphere Service Registry and Repository connection is available, the person publishing selects the appropriate option.

Deploy: retrieve deployed assets

After the service asset artifacts (WSDL, XSD, XML) are published to WebSphere Service Registry and Repository software and the repositories are synchronized, a shadow service asset is available in Rational Asset Manager software to support search and retrieval scenarios. In Rational Asset Manager software, the original service asset and the shadow service asset point to each other through an asset relationship.

The developer can search the service asset and the service shadow asset metadata and content. A link is also stored in the original service asset (in this case, the service interface asset and the deployable service asset) and the service shadow assets pointing to the service documents in WebSphere Service Registry and Repository software. This link is navigable, allowing the developer to go from Rational Asset Manager to WebSphere Service Registry and Repository software for further browsing, such as information that IBM Tivoli Composite Application Management software publishes into WebSphere Service Registry and Repository software.

There is one shadow service asset created for each of the documents published to WebSphere Service Registry and Repository software, as shown in figure 17.



Administration >

Repository Metrics

Number of assets: 42

Number of downloads: 10

Number of searches: 11

Reports

Repository Contents

Downloads

Search History

▿ Assets by group

Group

🟥 Delta Project Team
🟦 Mortgage Financing Team
🟨 RAM Group
🟥 RAM Modelers Group
🟧 Team A Analysts
⬛ Web Developers Team

Group	
DeltaProjectTeam	1
MortgageFinancingTeam	4
RAMGroup	12
RAMModelersGroup	15
TeamAAnalysts	8
WebDevelopersTeam	2

▹ Assets by type

▿ Assets by state

State

🟥 Review
🟦 Approved
🟨 Plan Review
🟥 Submitted

State	
Review	3
Approved	37
PlanReview	1
Submitted	1

Figure 17: Metrics and reporting

Providing searching and indexing at the service asset level and the service document level provides the developer with the granularity to work with specific service document levels, as well as the abstraction to understand related assets such as service tests and business processes.

Measure: measuring effectiveness

The next scenario is an example of the importance of a feedback loop within the SOA governance mechanism. A large telecommunications organization embraced an SOA strategy to improve staff productivity and lower costs. In keeping with that strategy, the commercial division development team made existing functionality available as services. However, no processes were put in place to measure the SOA's effectiveness in achieving specific business goals.

Rational Asset Manager software includes comprehensive metrics and reporting functionality to provide feedback to administrators. Examples of metrics and reports provided by the Rational Asset Manager application include:

- *Asset-level reports*
- *Downloads, browses, feedback and ratings*
- *Repository-level reports*
- *Assets by community, type and state*
- *Search history*
- *Downloads*

Conclusion

SOA governance is a critical part of the SOA puzzle, and IBM Rational has worked to address some of the challenges around service lifecycle management with the introduction of Rational Asset Manager software.

The Rational Asset Manager tool helps enable organizations to define, categorize, create, manage and govern assets as part of the service lifecycle. In conjunction with other parts of the IBM Rational Software Delivery Platform, including Rational ClearQuest and Rational ClearCase software, and with tight integrations with other IBM SOA solutions, such as WebSphere Service Registry and Repository software, the Rational Asset Manager tool can complete a full-service lifecycle solution that spans both development-time and run-time service assets.

For more information

To learn more about IBM Rational Asset Manger software, contact your IBM representative or visit:

ibm.com/software/awdtools/ram

To learn more about IBM Rational solutions for SOA governance, visit:

ibm.com/software/solutions/soa/gov

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
08-07
All Rights Reserved

The information contained in this documentation is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, it is provided "as is" without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this documentation or any other documentation. Nothing contained in this documentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM (or its suppliers or licensors), or altering the terms and conditions of the applicable license agreement governing the use of IBM software.