

# HP SOA Systinet

Software Version: 3.20

---

## Concepts Guide

Document Release Date: July 2009  
Software Release Date: July 2009



## Legal Notices

### *Warranty*

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

### *Restricted Rights Legend*

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

### *Copyright Notices*

© Copyright 2003-2009 Hewlett-Packard Development Company, L.P.

---

# Contents

About this Guide. . . . .	5
In this Guide. . . . .	5
Document Conventions. . . . .	6
Documentation Updates. . . . .	6
Support. . . . .	7
<b>1 SOA and SOA Systinet. . . . .</b>	<b>9</b>
<b>2 User Roles and the User Interface. . . . .</b>	<b>11</b>
Role-Specific User Interfaces. . . . .	11
Standard User Interface. . . . .	13
Graphical Navigator. . . . .	14
<b>3 SOA Definition Model. . . . .</b>	<b>19</b>
Artifacts. . . . .	19
Properties. . . . .	20
Taxonomies. . . . .	20
<b>4 SOA Repository. . . . .</b>	<b>23</b>
<b>5 Services. . . . .</b>	<b>25</b>
Service Implementation. . . . .	26
Environments. . . . .	28
Service Discovery. . . . .	29
Contracts. . . . .	30
<b>6 Governance. . . . .</b>	<b>33</b>
Lifecycle Overview. . . . .	33
Policy Overview. . . . .	45
Tools, Tasks, and Reports. . . . .	52

Versioning Overview.....	54
7 SOA Systinet Features.....	57
8 HP SOA Product Family.....	61

---

# About this Guide

Welcome to HP SOA Systinet, the foundation of Service Oriented Architecture, providing an enterprise with a single place to organize, understand, and manage information in its SOA. The standards-based architecture of SOA Systinet maximizes interoperability with other SOA products.



HP Software controls access to components of SOA Systinet with a license. This document describes the full functionality of SOA Systinet including licensed components. If your license does not include these licensed components, their features are not available.

## In this Guide

This guide provides an overview and introduction to the concepts used by SOA Systinet to manage an SOA instance.

This guide includes the following chapters:

- Chapter 1, SOA and SOA Systinet
- Chapter 2, User Roles and the User Interface
- Chapter 3, SOA Definition Model
- Chapter 4, SOA Repository
- Chapter 5, Services
- Chapter 6, Governance
- Chapter 7, SOA Systinet Features
- Chapter 8, HP SOA Product Family

## Document Conventions

This document uses the following typographical conventions:

<b>run.bat make</b>	Script name or other executable command plus mandatory arguments.
<code>[-help]</code>	Command-line option.
either   or	Choice of arguments.
<i>replace_value</i>	Command-line argument that should be replaced with an actual value.
{arg1   arg2}	Choice between two command-line arguments where one or the other is mandatory.
<code>java -jar hpsystinet.jar</code>	User input.
<code>C:\System.ini</code>	File names, directory names, paths, and package names.
<code>a.append(b);</code>	Program source code.
<code>server.Version</code>	Inline Java class name.
<code>getVersion()</code>	Inline Java method name.
<b>Shift+N</b>	Combination of keystrokes.
<b>Service View</b>	Label, word, or phrase in a GUI window, often clickable.
<b>OK</b>	Button in a user interface.
<b>New→Service</b>	Menu option.

## Documentation Updates

This guide's title page contains the following identifying information:

- Software version number, which indicates the software version
- Document release date, which changes each time the document is updated
- Software release date, which indicates the release date of this version of the software

To check for recent updates, or to verify that you are using the most recent edition of a document, go to:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport logon page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. For details, contact your HP sales representative.

## Support

You can visit the HP Software Support Web site at:

<http://www.hp.com/go/hpsoftwaresupport>

HP Software Support Online provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the HP Software Support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To find more information about access levels, go to:

[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)

To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

# 1 SOA and SOA Systinet

*Service-Oriented Architecture (SOA)* is an architectural style that defines the principles of service oriented architecture and programming. Service-oriented development focuses on systems that are built from a set of autonomous services. Therefore, SOA is about designing business functionality as shared reusable services.

SOA Systinet represents SOA in an extensible information model called the *SOA Definition Model (SDM)*. The SDM uses a declarative modeling language to describe SOA components. It defines SOA artifacts and the relationships between them.

The SDM enables you to model artifacts such as services and their relationships, metadata, and constraints.

You can then view and work with the artifacts in SOA Systinet as follows:

- **Repository**

A repository of key SOA artifacts including services and their relationships. SOA Systinet provides service providers and consumers with a common view of *business services* they offer and consume.

- **Discovery**

SOA Systinet indexes published artifacts so that you can be browse and search them using both full text search and query languages.

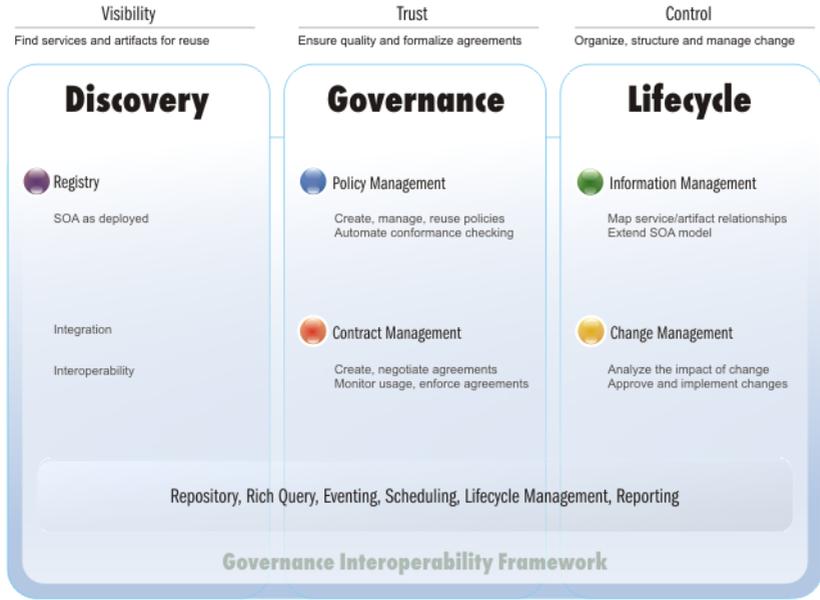
- **Full history**

SOA Systinet manages and maintains versions of all data published to the repository so you can track changes made to an artifact (for example, how a document evolved over time), see who modified it and when, and roll back a change if necessary.

- **Self-describing services**

When SOA Systinet contains an artifact, and represents its relationship to other artifacts, it is self-describing. In the case of service artifacts, this self-description provides service consumers with enough information to search and identify the services they choose to use.

**Figure 1. SOA Systinet Functional Overview**



After SOA infrastructure becomes an important part of enterprise internal systems, its functionality is critical. The blueprint of SOA artifacts (providing visibility, discovery, and reuse) is no longer sufficient. Over time, the number of services grows, and the relationships and dependencies between these services become more complex. You must control this growth within SOA. Only then can the agile growth of the SOA system be ensured, preventing redundant functionality, ensuring compliance with policies (both technical and enterprise one), and providing overall visibility.

SOA Systinet resolves this problem. It is a governance platform used to manage vital SOA.

## 2 User Roles and the User Interface

HP SOA Systinet is designed to be flexible. It supports a variety of typical requirements, yet recognizes that there are many other use cases and scenarios in the SOAs of different enterprises.

SOA Systinet provides a different user interface depending on your user role. There are a number of role-specific user interfaces with functionality targeted towards performing these roles and a standard user interface giving access to the full set of functionality for all user roles.

Roles are assigned by the administrator using group membership. For details, see "Managing Users" in the *HP SOA Systinet Administration Guide*.

This chapter describes these roles and user interfaces in the following sections:

- [Role-Specific User Interfaces on page 11](#)
- [Standard User Interface on page 13](#)
- [Graphical Navigator on page 14](#)

### Role-Specific User Interfaces

Role-specific user interfaces are provided for the following roles:

- **Business Partner**

Business Partner is a consumer role. They could be an external B2B user, group, or a company. Typically, service consumers are seeking to reduce time to market and reduce the cost of development. They build composite applications by searching for existing services, and negotiating contracts with service providers. The primary focus for this user interface is search and browse of consumable data, and the capability to create contracts.

Typical Business Partner use cases include the following:

- Discover Services for Reuse

Browses and search the repository for existing services to reuse.

- Request Consumption of a Service

When a service with the required functionality already exists then a Business Partner can initiate a contract to reuse the service by requesting consumption of the service.

Business Partners are only interested in consumable services. Often, they are external personnel only interested in last approved revisions of governed services or last revisions of ungoverned services.

For more details, see the .

- **Business Analyst**

A Business Analyst is a negotiator between the business side of an enterprise and the providers of services to the enterprise. The business analyst understands business problems and opportunities in the context of the requirements and recommends solutions that enable the organization to achieve its goals. Solutions often include a systems development component, but may also consist of process improvement or organizational change.

Typical Business Analyst use cases include the following:

- Search and Browse

Business Analysts browse and search the repository in order to find out where to invest in new services.

- Propose New Functionality

When Business Analysts cannot find required functionality (service, business process, or application) they propose new functionality by creating new business service, business process, or application artifacts.

- Attach Business Requirements

Business Analysts attach business requirements to newly created artifacts by uploading business requirement documentation.

- Request Approval

Business Analysts request approval for their proposed functionality.



For simplicity, HP Software refer to people who perform business analysis as Business Analysts. In many organizations, this role is referred to as Business System Analyst, System Analyst, etc.

Role-specific user interfaces are customizable during runtime, without the requiring redeployment of the EAR file or a restart of the SOA Systinet server. For details, see "Customizing the User Interface" in the *HP SOA Systinet Administration Guide*.

## Standard User Interface

To meet the needs of all other users, SOA Systinet provides a standard user interface with access to the full set of functionality.

The functionality set in the standard UI is controlled using *perspectives*. These perspectives are designed to meet the needs of typical user roles.



This documentation also uses these perspectives to explain SOA processes. Like other aspects of the system, perspectives are customizable.

The UI enables information to be viewed from any perspective based on permissions granted by the Administrator. Access to available information is limited only by security mechanisms.

Perspectives make it possible to present the information most relevant to particular users according to their typical use of SOA Systinet.

The following default perspectives are available in the standard UI:

- **Administrator**

The Administrator perspective is designed for the administration of users, groups, and perspectives in SOA Systinet as well as other critical functions. This perspective displays account management, access rights, and other features not intended for use by other users.



The Administrator role is a role in SOA governance and lifecycle management. The role is not necessarily part of the system administration of SOA Systinet itself.

- **Service Publisher**

The service publisher perspective is designed for users who publish, implement, and manage business services. The perspective displays service management and publication features not visible in the General perspective.

- **General**

The general perspective is intended for users who are not involved in the administration of SOA Systinet or the publication and implementation of services. Typically, they are service consumers who do not need to see service publication and management features.

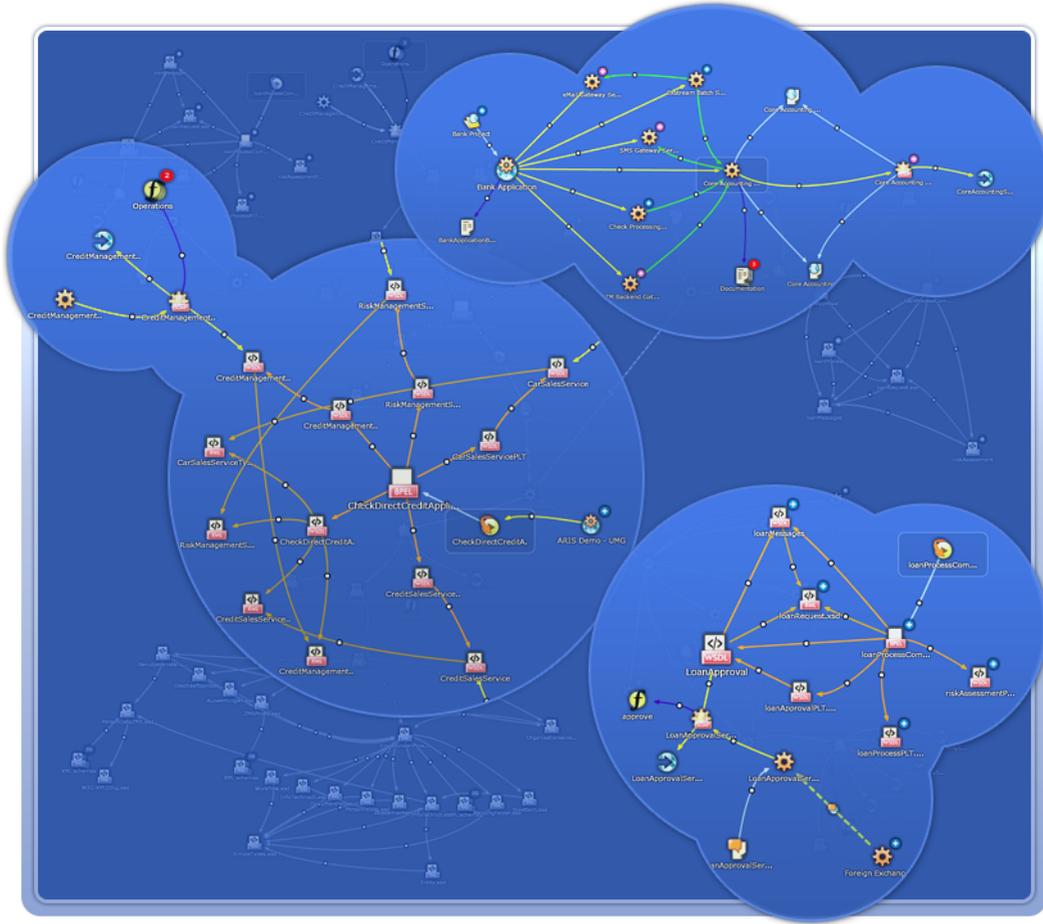
Administrators can manage the membership of perspectives in SOA Systinet. For details, see "Managing User Perspectives" in the *HP SOA Systinet Administrator Guide*.

You can configure existing perspectives and create additional ones using HP SOA Systinet Customization Editor. For details, see the *HP SOA Systinet Customization Editor Guide*.

## Graphical Navigator

The graphical navigator provides an alternative role-based view of the repository. Instead of focusing on the details of single artifacts, it helps to understand the structure and dependencies of applications, services, and processes modelled in your SOA repository. As its name suggests, it is not only an alternative view but also a navigator that enables the exploration, browsing, and traversing of your repository content in a graphical way. Simply put, the navigator is a graphical map and navigator for your SOA landscape.

**Figure 2. SOA Navigator Images**



Your SOA architecture can quickly become extremely complex, it is essential to have the right tools to help you to understand, present, discuss, and optimize your SOA.

For example, consider a Banking Application that consists of the following:

- Six interrelated banking services.

- Application documentation.

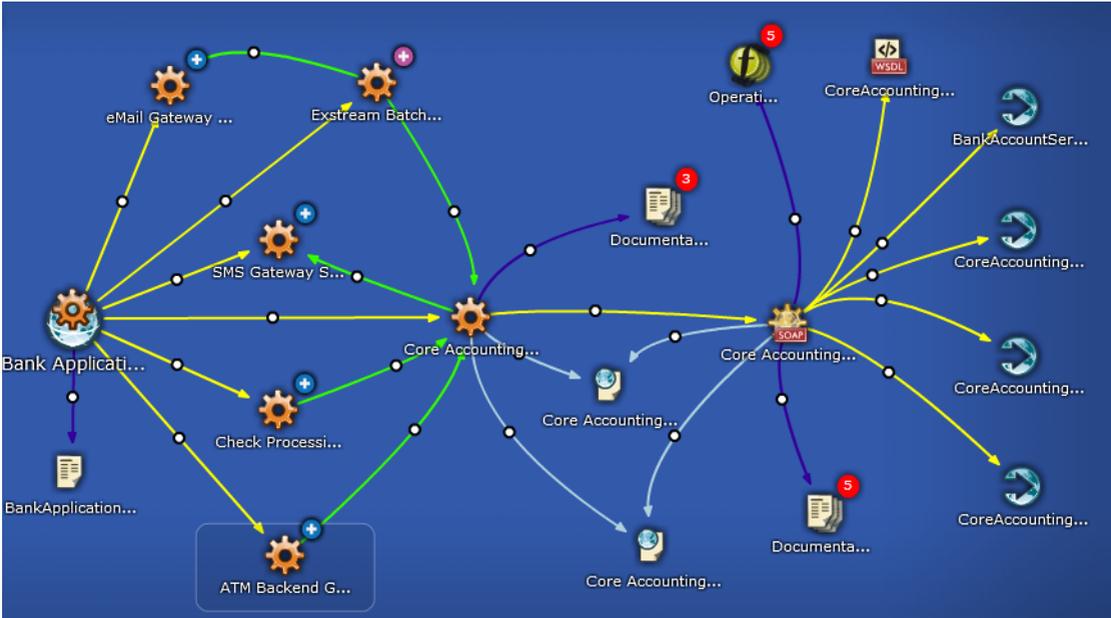
Each service can be examined in more depth. For example, the Core Accounting Service consists of the following:

- Three documents about the service.
- A SOAP Service implementation.
- A WSDL defining the implementation.
- Five operations of the implementation defined by the WSDL.
- Four endpoints of the implementation defined by the WSDL.
- Five documents about the implementation.
- Two sets of service level objectives which are applicable to either the service or the implementation.

This description is rapidly becoming more and more complex, even though it is only the structure of a single application and one of its services.

The Graphical Navigator provides an interactive overview of this complexity in a single image.

**Figure 3. Banking Application**



The main features of the Graphical Navigator are:

- **Sophisticated Layout Algorithms**

The navigator renders graphs automatically in an intelligible way. The graphs scale from small graphs of a few artifacts to large graphs depicting dependencies between thousands of WSDLs and XSDs.

- **Repository Browsing**

Simple and rapid navigation of repository content.

- **Perspectives**

The navigator provides a set of role-based perspectives focusing on the artifacts of interest to each role, as well as the capability to create custom perspectives.

- **Bookmarks**

The graph is interactive, enabling you to modify its appearance by moving, expanding, and collapsing nodes. You can add a particular graph layout to your bookmarks, enabling you to return at a later time to a particular layout for a particular graph.

- **Impact Reports**

The navigator provides a visual representation of impact and dependencies between artifacts.

- **Screenshots**

Take a screenshot of particular graph layout.

The navigator can also help collaboration between SOA Systinet users. For example, an architect can demonstrate the structure of an application to the developers who implement it, and the developer can explain dependencies the application has on corporate schemas to QA engineers.

The main benefits of the navigator vary according to the role of the user:

- *Architects* can analyze the structure and dependencies of applications, services, and business processes.
- *Developers* can quickly understand the complex structure of service interfaces defined by interconnected WSDLs and XSDs, and understand the business context of service implementations.
- *QA Engineers* can understand the context and structure of dependencies for tested services and review the impact of any changes in order to focus only on those services affected by a change.
- *Operation Engineers* can assess the purpose and business context of services and applications.
- *Managers* can use the navigator to improve communication with architects and business analysts.

For details of the functionality of the Graphical Navigator, see the "Graphical Navigator" section in the *HP SOA Systinet User Guide*.

## 3 SOA Definition Model

At the core of HP SOA Systinet is the SOA Definition Model (SDM). The SDM enables you to organize, understand, and manage the business, policy, contract, and deployment data relating to SOA services, categorizations, relationships, and specifications. The SDM defines the SOA Systinet data schema, reflecting the actual semantics of the SOA in your enterprise. As a result it determines the structure of data in the repository described in the *HP SOA Systinet Reference Guide*.

The flexibility and extensibility of the SDM ensure that SOA Systinet can keep up with the evolution of your SOA.



You can customize the SDM to meet your requirements using the HP SOA Systinet Customization Editor. For details, see the *HP SOA Systinet Customization Editor Guide*.

This chapter introduces the basic concepts of the SDM in the following sections:

- [Artifacts on page 19](#)
- [Properties on page 20](#)
- [Taxonomies on page 20](#)

### Artifacts

SOA Systinet enables you to describe your SOA with the following attributes:

- **Metadata**  
Attributes that describe an artifact and its relationships to other artifacts.
- **Abstract**

Artifacts that do not have actual instances in the repository. These artifacts are the parent types of other artifact types that inherit their properties.

- **Instances**

Artifacts in the repository. Each artifact is an instance of an artifact type.

The artifact types form a hierarchical taxonomy, described in detail in "SOA Definition Model" in the *HP SOA Systinet Reference Guide*. Each branch in the tree of artifact types represents a category corresponding to a single type. The purpose of other artifact types is to represent categories corresponding to a number of artifact types. The root type artifact represents a category containing all SOA artifact types defined.

## Properties

Artifacts have properties that define specific attributes of an artifact instance.

SOA Systinet uses the following property types:

- **Primitive Properties**

Examples of primitive properties are name, description, and timestamp.

- **Taxonomy-based properties**

Taxonomies are used for the categorization of artifacts. Examples include geographical location, lifecycle stage, and importance.

- **Relationships**

Relationships capture logical as well as physical dependencies between artifacts (for example, enabling impact and what-if analysis). Relationship-based properties are either explicitly created by users or implied by the model for a type of source document and related target document.

## Taxonomies

Taxonomies are also known as value sets. A taxonomy defines a set of values that can be used for the categorization of artifacts.

SOA Systinet typically uses taxonomies to define sets of fixed values. These are usually referred to as categories. The categories of a taxonomy can be organized either hierarchically or as a flat list.

In SOA Systinet, taxonomies are not checked, which means that property values (categories) are not validated.

You can customize the set of taxonomies defined in SOA Systinet. For details, see the *HP SOA Systinet Taxonomy Editor Guide*.



---

## 4 SOA Repository

The repository is the content management system for HP SOA Systinet, reflecting the SOA Definition Model. As a result the repository can be considered the foundation of SOA Systinet. Users access the repository indirectly through higher level features described in the *HP SOA Systinet User Guide*.

The repository is a database designed to store documents. It has special features for accessing XML document content. Whereas relational databases are designed to hold tabular data, the repository supports the hierarchical organization of documents and the data they contain. Features of the repository include versioning, document management, reliability, and failover.

Complementary to the repository is the reporting service, which is a customizable, extensible query component enabling you to access the data you need quickly and efficiently.

The repository includes the following resource types:

- **Documents**

Similar to files or rows in a relational database. Each document can include SDM metadata and relationships with other documents.

- **Collections**

Similar to directories or tables in a relational database. Each collection can contain other collections. These collections form a hierarchy.

For details about the structure of the repository and the format of the resources it contains, see the *HP SOA Systinet Reference Guide*.



---

## 5 Services

HP SOA Systinet uses a hierarchy of service artifacts that enable you to manage your SOA at a level suitable to your organization.

The following service artifact types are defined:

- **Application**

The application artifact enables you to organize your SOA at a high level.

Typically, an application contains a set of business services.

- **Business Service**

The business service is the central element in SOA.

Typically, a business service contains an implementation.

- **Implementation**

The implementation represents the functional element of a business service.

SOA Systinet supports the following implementation types:

- SOAP Service
- XML Service
- Web Application

Implementations are typically associated with endpoints or operations.

- **Endpoint**

Endpoints represent URLs used to access a service implementation. Throughout the lifecycle of a service, there are likely to be a number of different environments hosting a service, for example, development, testing, and production servers. Use a different endpoint to represent each of these environment implementations.

- **Operation**

An operation represents a method defined in an implementation.

SOA Systinet includes the Service Catalog to publish your infrastructure service and then make it available to consumers, as described in the following sections:

- [Service Implementation on page 26](#)

SOA Systinet enables you to create a service infrastructure in the repository.

- [Environments on page 28](#)

SOA Systinet enables you to apply an environment taxonomy to endpoints and integrated product servers simplifying service lifecycle management.

- [Service Discovery on page 29](#)

SOA Systinet enables you to publish service infrastructure from external sources.

- [Contracts on page 30](#)

SOA Systinet enables you to establish contracts with associated service level objectives.

## Service Implementation

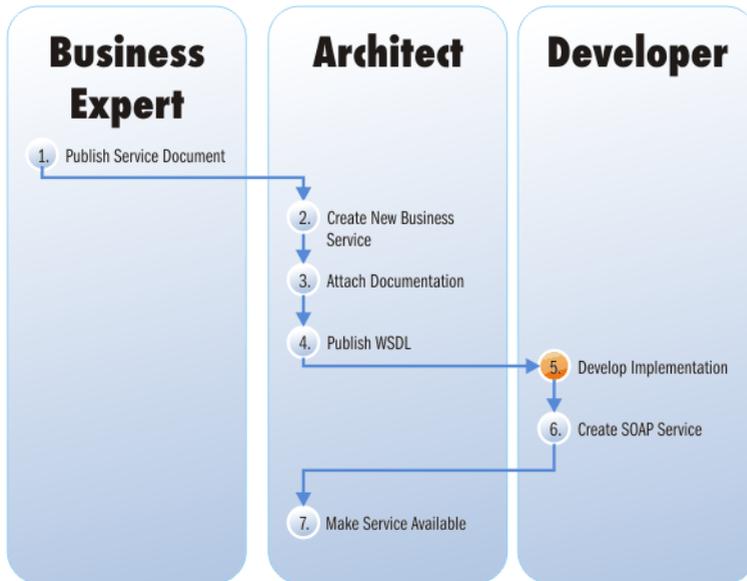
One of the main uses of SOA Systinet is to publish and implement business services.

SOA Systinet uses a set of publication wizards, enabling you to add services to your SOA and make them available to consumers.

Figure 4, “[Business Service Workflow](#)” describes a typical publication process and the steps required to achieve it. The workflow is split into the user roles that would typically perform the actions in an organization

using SOA. Each step is described in brief in this section and explained in detail in the *HP SOA Systinet User Guide*.

**Figure 4. Business Service Workflow**



**Key to Figure 4, “Business Service Workflow”:**

- 1 Business expert publishes a service document in SOA Systinet describing a proposed new service.
- 2 Architect receives notification of the new service requirement, reviews the documentation, and creates a new business service.
- 3 Architect creates a new technical document. Architect attaches the new technical document and the existing service document to the business service. to the service.
- 4 Architect publishes a WSDL document, which contains a technical description of the service.
- 5 Developer reviews the documentation and uses the WSDL to develop an implementation of the business service (for example, in Microsoft Visual Studio).

- 6 Developer adds a SOAP service is added to the business service, with all the relevant documentation.
- 7 Architect makes the business service available by editing it, changing its Lifecycle Stage to production, and setting the Enable Consumption Requests flag.



This can be automated as part of a lifecycle governance process.

## Environments

Most organizations use several environments to clearly separate production from testing and development. This means that when a service moves through its lifecycle, it is likely to be deployed multiple times in a number of different environments.

SOA Systinet makes use of an Environment taxonomy to represent these environments. Each environment is represented in SOA Systinet by an environment category which can be applied to endpoints and server artifacts for other products integrated with SOA Systinet (for example, UDDI and BAC servers).

This categorization ensures that only the appropriate information is automatically exchanged between SOA Systinet and integrated servers.

This approach avoids the constant republishing of WSDLs defining services to reflect these additional endpoints as the service moves through its lifecycle. SOA Systinet provides environment specific WSDLs that automatically contain the appropriate endpoints whenever a WSDL is required.

A typical example for a service could be described as follows:

### 1 Development

A service is likely to be defined by publishing a WSDL before it reaches the development stage. When it reaches the development stage it requires an endpoint representing a development server. Instead of amending and republishing the WSDL, SOA Systinet enables you to add an endpoint categorize it as Development.

### 2 Testing

During the testing stage, the service may require an additional endpoint representing test deployment of the service and this is also achieved by adding a new endpoint categorized as `Testing`.

SOA Systinet may be integrated with *Service Test Manager (STM)*. When a service is registered in STM, the associated WSDL for the test environment is exported to STM. By categorizing the STM server as `Testing`, you ensure that only the Testing endpoints are visible to the STM server.

### 3 Production

Similarly in production, the service requires an additional endpoint representing the production deployment of the service and this is also achieved by adding a new endpoint categorized as `Production`.

When a service reaches production you may wish to export it to a UDDI registry. By categorizing the registry server as `Production` you ensure that the service is exported with the appropriate production endpoints.

## Service Discovery

HP SOA Systinet offers publication methods, enabling you to import service infrastructure from external sources.

The following methods are available:

- **Publishing Services**

SOA Systinet includes a publisher that creates a service infrastructure defined by a service definition document, such as WSDL.

- **Importing Services from UDDI Registries**

SOA Systinet integrates with UDDI Registries, enabling you to use a registry as a central point for several products to share service infrastructure.

- **Service Discovery from UCMDB**

SOA Systinet integrates with the *Universal Configuration Management Database* used by *HP Business Availability Center*.

For details, see "Service Discovery" in the *HP SOA Systinet User Guide*.

## Contracts

[Service Implementation on page 26](#) describes the process of populating SOA Systinet with services. A key service relationship is the provider-consumer relationship that you can create between service consumers and providers.

There are several reasons to capture provider-consumer relationships in SOA Systinet:

- **Change Management**

By requiring consumers of services to register, the provider ensures that all users of that service are notified whenever some aspect of it changes.

- **Consumption Access Control**

A controlled way of providing consumers access to a service, helping prevent unanticipated or inappropriate usage.

- **Consumption Monitoring**

Monitors how successful a particular service is, based on its level of reuse and its use across organizational units.

- **Consumption Auditing**

Providers need to know who is using specific services and who granted them access. This could be for many reasons, such as security, accounting, capacity planning, or change management.

SOA Systinet provides Contract Manager to facilitate these functions. Contract Manager enables you to specify conditions under which the provided service can be consumed, as service level objectives.

Consumers can request the consumption of a particular provider by choosing one of the service-level objectives provided. You then establish a contract with the consumer, describing the service provided, and under what conditions it can be consumed.

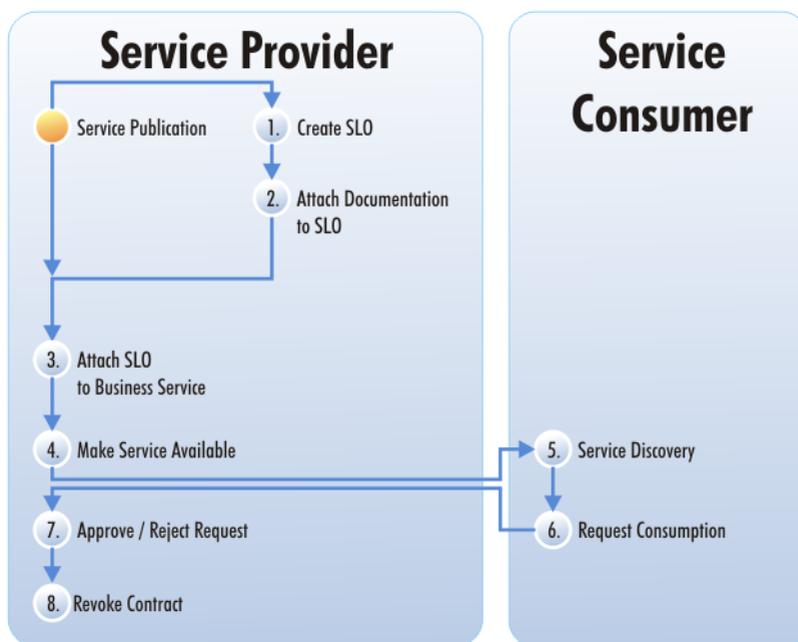
For a provided service to be consumed, there are several steps required by the provider and a discovery process for the consumer. Having found a service that meets their requirements the consumer and provider can establish a contract. The [Figure 5, “Contract Workflow”](#) displays the typical tasks that should be performed, and an overview of each step is given below.



The workflow uses a business service as the example. SOA Systinet enables other artifact types to be providers as well. For details, see "Managing Contracts" in the *HP SOA Systinet User Guide*.

The procedure for each step is described in detail in the *HP SOA Systinet User Guide*.

**Figure 5. Contract Workflow**



**Key to Figure 5, "Contract Workflow":**

- 1 After you create a service, it is normal practice to offer a number of service-level objectives describing the terms of use for the service. You can create an SLO independently, or as part of a service.
- 2 You can also attach documentation to the SLO.

- 3 Attach the SLO to the Business Service.
- 4 Make the Business Service available by marking it as *consumable*.



This can be automated as part of a lifecycle governance process.

- 5 The consumer browses SOA Systinet to see which services are available and whether they meet their requirements. See *Exploring the Repository*.
- 6 After the consumer finds the service, they can request it. See *Requesting Consumption*.
- 7 When a consumer requests consumption of a service, you must either approve or reject the request. See *Processing Consumption Requests*.
- 8 You can also cancel an existing contract. See *Revoking an Active Contract*.

---

# 6 Governance

HP SOA Systinet provides a full suite of governance utilities enabling you to control your SOA as it grows and evolves.

This chapter introduces the key concepts in SOA Systinet governance:

- [Lifecycle Overview on page 33](#)

An introduction to lifecycle governance in SOA Systinet.

- [Policy Overview on page 45](#)

Policy Management enables you to ensure that your production takes place in conformance with your business and technical guidelines.

- [Tools, Tasks, and Reports on page 52](#)

The SOA Systinet suite of tools enables you to create a set of automated scheduled tasks that produce reports giving you complete visibility and control of your SOA.

- [Versioning Overview on page 54](#)

An introduction to version management in SOA Systinet.

## Lifecycle Overview

SOA Systinet enables you to apply a specified development path to a set of service artifacts.

A governance process consist of the following elements:

- [Lifecycle Stages and Transitions on page 34](#)
- [Default Governance Processes on page 36](#)

- Governed Artifacts on page 41
- Stage Approval Conditions on page 44
- Automatic Actions on page 44

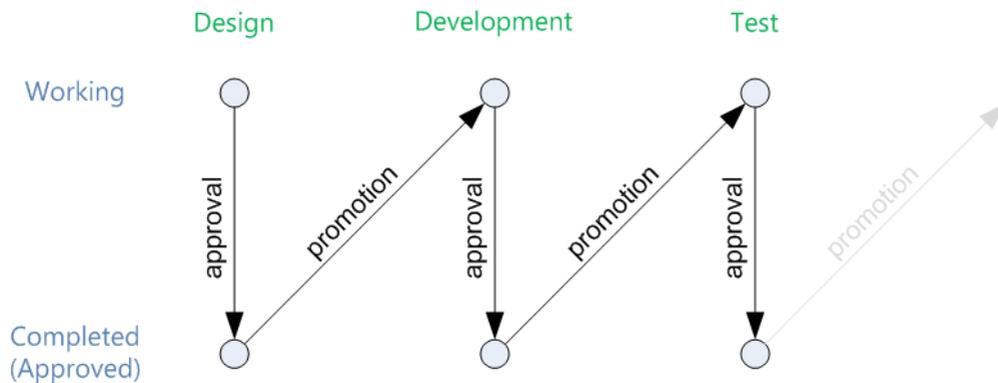
For full details about setting up and using lifecycle, see "Lifecycle Governance" in the *HP SOA Systinet User Guide*.

## Lifecycle Stages and Transitions

A lifecycle consists of a set of stages and transitions between them.

Figure 6, "Lifecycle Overview" shows part of a governance process.

**Figure 6. Lifecycle Overview**



The current stage of a service in a lifecycle can be described and queried in the following ways:

- **Working Stage**

The working stage represents the stage currently in progress.

- **Approved Stage**

The approved stage represents the last completed stage. Approval requires the completion of a set of conditions described in [Stage Approval Conditions on page 44](#).

Moving between stages involves the following lifecycle transitions:

- **Approval**

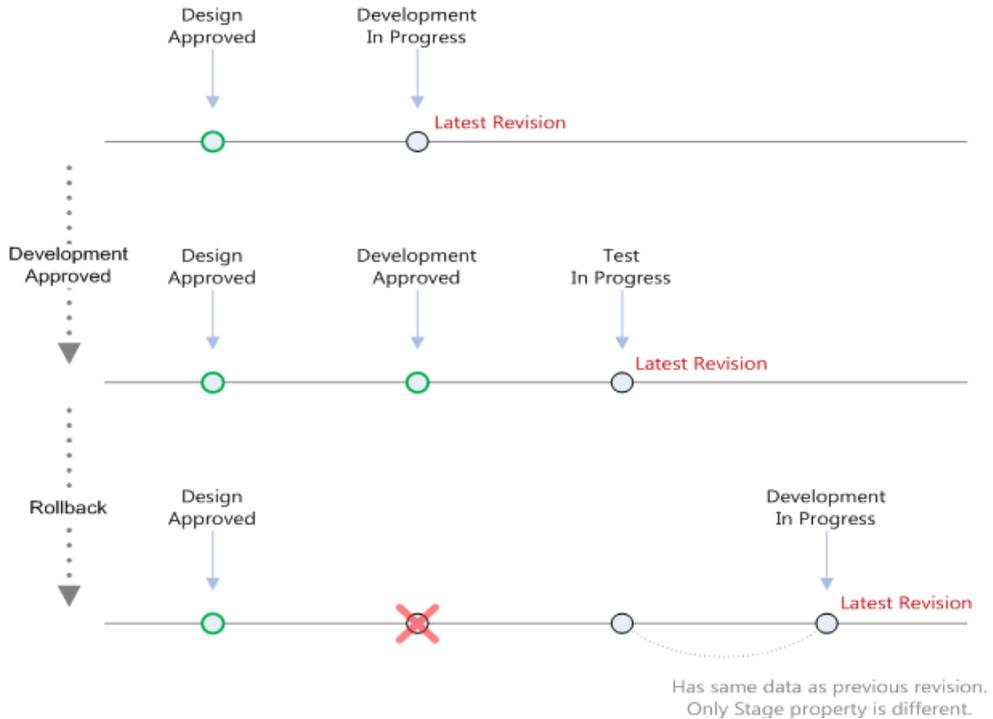
The approval process is the confirmation that all the requirements for the current Working Stage are complete and results in the Approved Stage.

- **Promotion**

Promotion initiates the next Working Stage. This requires the current Working Stage to be Approved and can be automated as part of the process definition.

- **Rollback**

You can reverse a lifecycle stage approval subject to rollback approval voting. This may result in a demotion of the Working Stage as well.



- **Demotion**

Demotion cancels the current Working Stage and returns to the previous one. This has no impact on the Approved Stage.

There may be a set of automatic actions associated with a lifecycle transition, as described in [Automatic Actions on page 44](#)

## Default Governance Processes

SOA Systinet provides a set of predefined governance processes. SOA Systinet automatically assigns newly created artifacts to these processes (unless another process is a better match, for details, see [Governed Artifacts on page 41](#)). These processes contain preset policies and tasks for each lifecycle stage and are

linked together by a set of technical policies (see "SOA Systinet Technical Policies" in the *HP SOA Systinet User Guide*).

The following sections describe these default lifecycles and their relations to each other:

- [Project Lifecycle on page 37](#)
- [Application Lifecycle on page 38](#)
- [Service Lifecycle on page 39](#)
- [Service Implementation Lifecycle on page 40](#)
- [Process Lifecycle on page 41](#)
- [Process Implementation Lifecycle on page 41](#)

## Project Lifecycle

Projects represent any progressive activity (e.g. service development). Projects can be composed of services, application, business processes, and other artifacts. In this release, the goal of a project is to successfully deploy its constituent artifacts to a production environment.

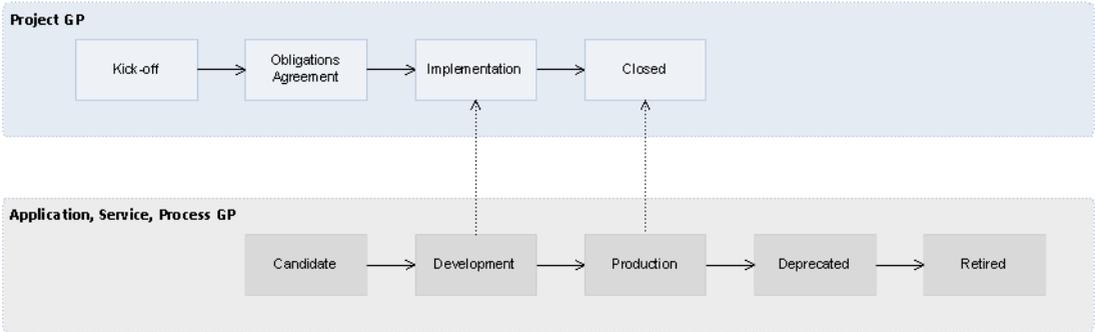
### **Project Governance Tree:**

- Project
  - Documentation

Promotion is automatic after the approval of each stage.

Figure 7, “[Project Lifecycle](#)” shows the Project Governance Process stages and its dependencies.

**Figure 7. Project Lifecycle**



The Project Governance Process depends on the governance processes of its constituent artifacts (applications, services, and business processes). For example, the Closed stage of a project cannot be approved until all its constituent artifacts are approved in the Production stage.

**Application Lifecycle**

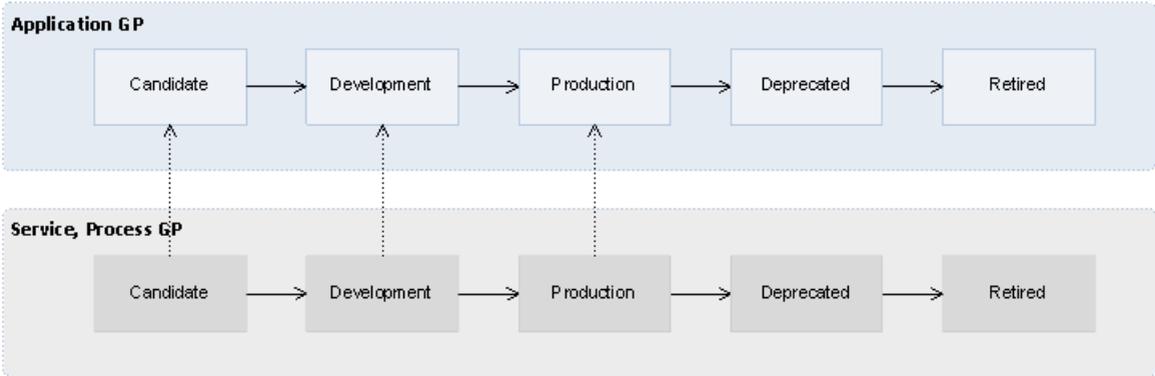
Applications represent business functionality in the real world. Applications can consist of multiple components (business services and business processes) but these do not form part of the application governance tree.

**Application Governance Tree:**

- Application
  - Documentation
  - SLO

Figure 8, “Application Lifecycle” shows the Application Governance Process stages and its dependencies.

**Figure 8. Application Lifecycle**



The Application Governance Process depends on the governance processes of its component artifacts. For example, the Production stage of an application cannot be approved until all its component artifacts are approved in the Production stage.

**Service Lifecycle**

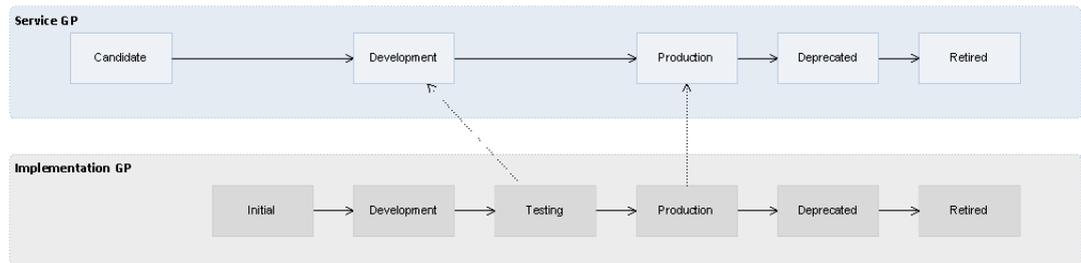
Services represent business functionality (or concepts) in the real world. Services consist of sub-services and implementations but these do not form part of the governance tree.

**Service Governance Tree:**

- Service
  - Documentation
  - SLO

Figure 9, “Service Lifecycle” shows the Service Governance Process stages and its dependency on the Implementation Governance Process.

**Figure 9. Service Lifecycle**



The Service Governance Process depends on the governance processes of its constituent artifacts. For example, the Development stage of a service cannot be approved until all its sub-services are approved in the Development stage and at least one of its implementations is approved in the Testing stage.

### Service Implementation Lifecycle

In the real world an implementation represents a specific service version being implemented, in the repository it represents a package with the service executables which can be later deployed to multiple environments.

### Service Governance Tree:

- Implementation
  - Operation
  - Endpoint
  - Interface (WSDL)
  - Schema (XSD)
  - Documentation
  - SLO

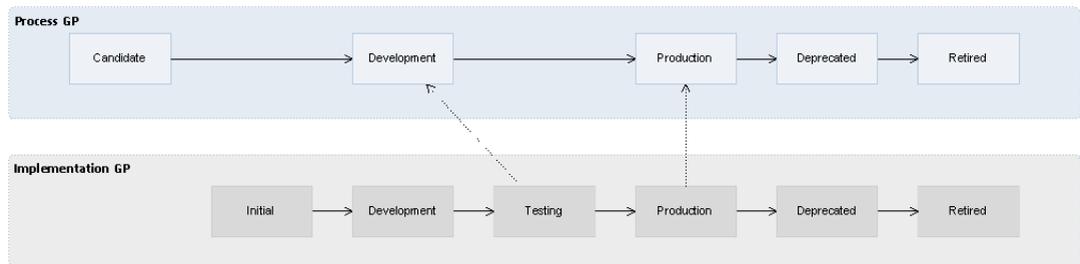
Each implementation represents one service and the Service Governance Process depends on the Implementation Governance Process. For details, see [Service Lifecycle on page 39](#).

## Process Lifecycle

The Process lifecycle is similar to the Service Lifecycle. Some of the tasks and policies vary, but the stages and dependency on the process implementation lifecycle is the same as that between a service and its implementation.

Figure 10, “Process Lifecycle” shows the Process Governance Process stages and its dependency on the Implementation Governance Process.

**Figure 10. Process Lifecycle**



## Process Implementation Lifecycle

The governance process for business process implementations is similar to that for service implementations. Some tasks and policies vary, but the stages and relationships to the process lifecycle are the same.

## Governed Artifacts

You can define a governance process to apply to *Governance Trees* which are hierarchies of artifacts.

The hierarchy is defined using *Root* and *Sub-Artifact* artifact types where sub-artifacts are usually related artifacts with an impact on a root artifact. For example, a Business Service and an Implementation of that service form a natural hierarchy that should be governed as part of the same process.

An artifact belongs to a particular governance process (and may be automatically added to it) when the following conditions apply:

- The artifact type is defined as a root or sub-artifact of a governance process.
- There is a relationship to the artifact from a governed artifact.

- The artifact is not governed by another process.

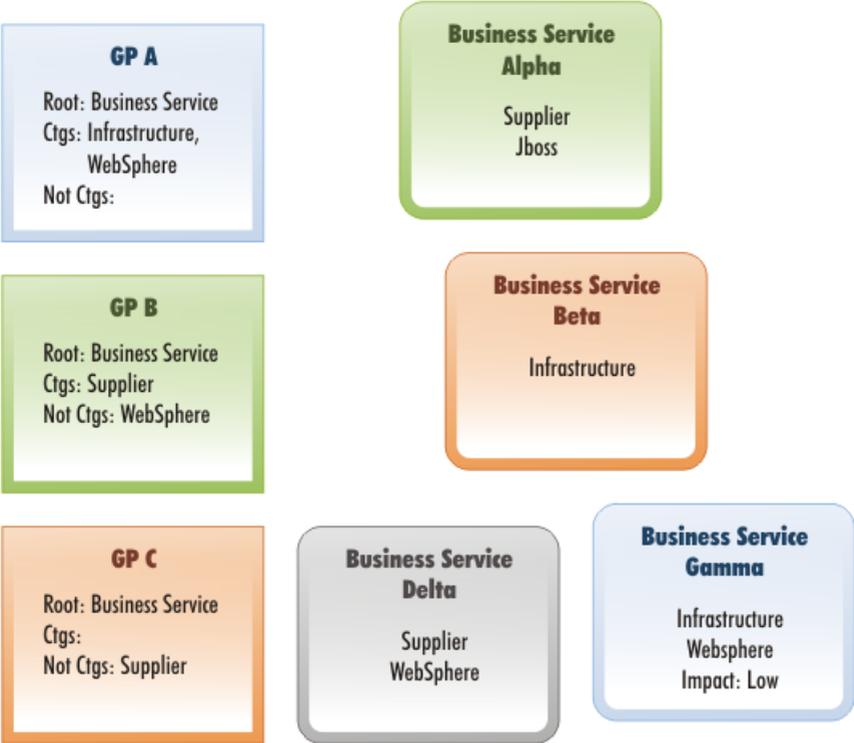
When an artifact is created, it may be automatically assigned to an appropriate governance process.

Automatic assignment works in the following way:

- 1 SOA Systinet checks that there are governance processes for the artifact type which are marked as **Automatically Assigned**.
- 2 If the governance process has taxonomic category conditions associated with it, SOA Systinet checks that the artifact matches these conditions. The following conditions apply:
  - The artifact type is a root artifact for the governance process.
  - **In Categories**: The artifact must contain these categories to be assigned to the governance process.
  - **Not In Categories**: The artifact must not contain these categories to be assigned to the governance process.
- 3 The artifact is assigned to a governance process as follows:
  - If there is no matching governance process, the artifact is not assigned to any governance process.
  - If there is one matching governance process, the artifact is assigned to it.
  - If there are two or more matching governance processes, the artifact is assigned to the most appropriate one based firstly on the number of matching **In Categories**, and then the number of matching **Not in Categories**.

Figure 11, “Automatic Assignment of Business Services” shows a number of governance processes with category associations and categorized business services.

**Figure 11. Automatic Assignment of Business Services**



- Business Service Alpha is assigned to Governance Process B because of the *Supplier* category match.
- Business Service Beta is assigned to Governance Process C. It cannot be assigned to Governance Process A because it is not categorized as *WebSphere*.
- Business Service Gamma is assigned to Governance Process A because of the *Infrastructure* and *WebSphere* category matches.

- Business Service Delta cannot be assigned to a governance process so it remains unassigned. Governance Process A specifies that an artifact must be **In Category Infrastructure** and Governance Processes B and C both specify **Not In Categories** used by the business service.

## Stage Approval Conditions

At each lifecycle stage in a governance process you can define the following governance conditions to help determine whether an artifact is ready for approval.

- **Policies**

You can associate technical policies with root or sub-artifacts. SOA Systinet automatically validates governed artifacts against these policies as part of the approval process.

- **Tasks**

You can associate manual tasks with root or sub-artifacts. These tasks can be tracked, requiring them to be marked as completed as part of the approval process.

- **Approvers**

At each lifecycle stage, you can define a set of approvers whose vote is required to make the current Working Stage into the current Approved Stage. These votes can be assigned to individuals or groups and may also be marked as required. You can also apply approval voting to Rollback transitions.

## Automatic Actions

At each lifecycle stage, you can define a set of *Automatic Actions* to modify the artifact when certain lifecycle transitions apply.

The available automatic actions are:

- **Change Access Rights**

Modify read and write permissions as a result of a lifecycle change. For example, give QA write access when an implementation is promoted to the Testing stage.

- **Notifications**

Send automated mail notifications to sets of users as a result of a lifecycle change. For example, notify consumers that a new version of a business service is approved in the Production stage and is available for consumption.

- **Enable/Disable Consumption**

Allow or disallow the creation of contracts for an artifact as a result of a lifecycle change. For example, set the Consumable flag when a Business Service in the Production stage is approved.

- **Change Owner**

Modify the user who owns an artifact as a result of a lifecycle change. For example, change the owner to an Architect when a business service is promoted to the Initial stage.

## Policy Overview

HP SOA Systinet Policy Manager provides an open and extensible framework that helps development and architecture teams achieve and maintain design consistency. It automates the validation of *SOA artifacts* and *registry* content for *compliance* with corporate guidelines and industry best practice.

This section describes the following aspects of policy management:

- [Policy Manager Usecases on page 46](#)

When and why you use Policy Manager

- [Policy Concepts on page 46](#)

The concepts Policy Manager uses.

- [Policy Workflow on page 48](#)

The lifecycle of policy development and report generation.

- [Policy Manager Components on page 50](#)

The technical components of Policy Manager.

## Policy Manager Usecases

Policy Manager has been designed for these primary use cases:

- **To establish and maintain consistent design of SOA artifacts**

Policy Manager automates validation of SOA artifacts (such as WSDL, XSD, or other XML documents) for policy compliance at design and testing stages.

- **To support and maintain the quality of SOA Systinet**

Policy Manager can check the contents of the SOA Systinet repository for their compliance with published policies.

## Policy Concepts

Policy Manager uses the following concepts in its model, from lowest to highest level:

- [Assertions on page 46](#)
- [Business and Technical Policies on page 47](#)
- [Compliance Validation and Reports on page 47](#)
- [Compliance Status on page 48](#)

### Assertions

The individual requirements that comprise a technical policy are called *assertions*. Assertions contain definitions of how validation issues are to be resolved. These definitions include a validation handler, which is a pluggable piece of code for validating the assertion.

The following validation handler dialects are supported:

- XPath
- XQuery
- Manual

Policy Manager enables you to enforce technical and business policies. These are associated with artifacts and can be checked for compliance either manually or on a scheduled periodic basis.

*Manual assertions* enable criteria to be specified that cannot be checked automatically. For example, "Every WSDL must be appropriately documented." These can only be enforced using manual compliance checks.

You can also implement your own validation handlers in Java, which can be added as extensions in HP SOA Systinet Assertion Editor.

## Business and Technical Policies

Policy Manager introduces the concept of *Business Policies*. A business policy is a set of requirements that apply to a specific set of artifacts. It should always represent a course of action that is needed to achieve a particular business objective.

For example, a bank may have a policy that all operations implemented as web services must be interoperable with external business entities. Therefore, their WSDLs must be well documented and conform to the bank's internal standards.

Every business policy consists of two major parts:

- Identifying a group of artifacts
- Sets of requirements that apply to the identified artifacts

In the Policy Manager model, the identification of artifacts is called a *Scope of Application* and the sets of requirements are named *Technical Policies*. In the example of the bank's business policy, its scope of application consists of all operations open to business partners that are implemented as web services. The technical policies being applied are the requirements that all WSDLs must have documentation and conform to the bank's internal standards.

## Compliance Validation and Reports

There are two types of compliance validation:

- **Business policy validation**

Artifacts identified in a business policy are checked for compliance with the individual assertions in the associated technical policies, making it easy for you to identify if the business objectives set forth in policies are being met.

- **Resource compliance**

A lower-level check of a single artifact against one or more technical policies. This feature is provided as a testing tool.

The results of both types of compliance checks are shown in *reports*. Reports are stored in the SOA Systinet repository.

The main function of Policy Manager is enabling you to check whether enterprise IT policies are being followed. This takes place when a business policy is validated. During validation, the artifacts associated with a business policy are checked for compliance with the assertions contained within that policy's technical policies. These validations can be run from within Policy Manager, from the SOA Systinet tools, or as prescheduled SOA Systinet tasks.

In the course of policy development, you might want to test an artifact by checking it against one or more technical policies. This kind of compliance check is also possible in Policy Manager. Compliance validation can be performed automatically or manually, depending on the type of assertion involved.

In addition, you can track the compliance of your SOA to your policy over periods of time using trend reports.

## Compliance Status

Compliance status is defined as the percentage of passed items from the set of technical and business policies applicable to an artifact. The compliance status for artifacts is recalculated by a nightly task or manual request.

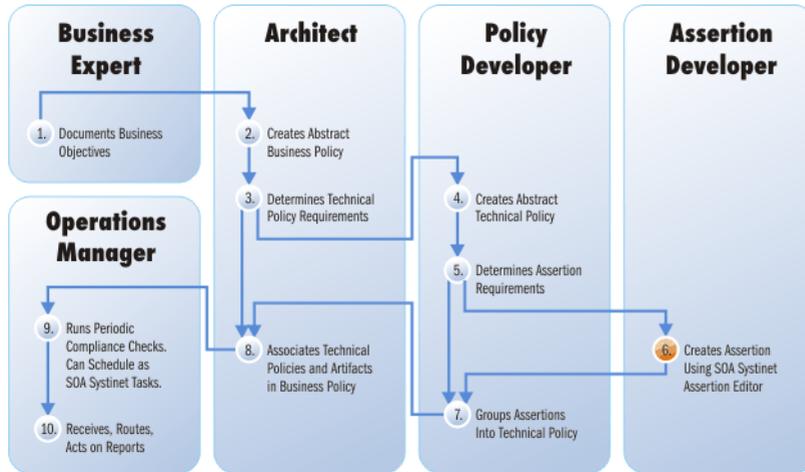
Compliance status is displayed in details and list views of these artifact types as a percentage.

The displayed compliance status may be obsolete if the artifact has changed since the compliance status was last calculated. In this case the compliance status is shown in parentheses.

## Policy Workflow

The workflow of policy creation and its subsequent use is shown in [Figure 12, “Policy Workflow”](#).

**Figure 12. Policy Workflow**



The policy workflow is broken down to the following steps:

- 1 The business expert documents business objectives in the SOA Systinet repository. The business expert then instructs the architect to create business policies fulfilling these objectives.
- 2 Using Policy Manager, the architect first creates an abstract business policy — with only the name and description — related to one or more business objectives.
- 3 The architect now determines what technical policies are needed for the business policy. If these technical policies already exist in the repository, the architect can move on to Step 8. Otherwise, the architect instructs the policy developer to create them.
- 4 The policy developer creates an abstract technical policy, with only the name and description.
- 5 The policy developer determines what assertions to implement in the technical policy. If all the assertions already exist in the repository, the policy developer can move on to Step 7. Otherwise, the policy developer instructs the assertion developer to create them.
- 6 The assertion developer creates, tests, and publishes the assertions needed by the policy developer, using HP SOA Systinet Assertion Editor.

For details, see the *HP SOA Systinet Assertion Editor Guide*.

- 7 The policy developer groups the assertions into technical policies, and then runs resource compliance checks to test the technical policies against artifacts of the relevant type, before publishing the policies to the SOA Systinet repository.
- 8 The architect associates technical policies with artifacts in the business policy, and then runs compliance checks to test the business policy before publishing it.
- 9 The operations manager can now use the business policy to enforce business objectives, and can run periodic compliance checks or schedule a SOA Systinet task to perform them automatically.
- 10 The operations manager acts on the reports generated by compliance checks.

## Policy Manager Components

Policy Manager includes the following components:

- **Policy UI**

This is a user interface (UI) for browsing, viewing, and editing business and technical policies. You can access the policy UI from a menu of links displayed in all policy pages and from related parts of the SOA Systinet UI.

The policy editor does the following:

- Uploads the policy to the UI session.
  - Enables the policy to be edited in the session.
  - Saves the policy in the SOA Systinet repository.
  - Enables the policy to be tested against a local or remote artifact.
- **Validation API**

Using this API, you can validate an artifact against a policy.

- **Remote Validation API**

This API is intended for remote validators. Using this API, a remote validator (either a service for automatic or a user for manual compliance checks) can asynchronously set the validation status of an assertion.

- **Report Visualizer/Editor**

This is functionally similar to the policy viewer/editor. It provides an editor for manual assertions contained in the report.

- **Task Queue**

A JMS queue for posting validation tasks. When a report is initiated, a task is put into the queue to start the validation process.

- **Validation Engine**

Engine for validating input documents against policies. The resulting report is stored in the SOA Systinet repository.

The engine also contains implementations for remote validators. At any time, a validator can send the validation status of a single assertion and the validation engine is responsible for adding this status to the report and recomputing the overall report status (running/not running, ok/error, and so on).

- **Validation Handler**

This is a pluggable piece of code whose task is to validate a single assertion (or a collection of assertions of the same type, in the case of WSI, XQuery, and so on.) against an input document. Its input consists of the input document, assertion being validated, definition of the assertion, and a set of credentials (if the handler needs to contact other artifacts which might be secured). Its output is a validation report for this single assertion (list of assertions).

A validation handler should never throw an exception. The rationale is that in case of a failure, the validation handler should return a meaningful report for each of the assertions, not a generic exception indicating that one or more assertions in the policy has failed.

A validation handler can be either local (implemented and running within the same JVM) or remote (for example, one of the runtime assertions implemented in runtime governance).

- **Source Loader**

This is a component for loading an object representation of XML source documents from a URL or DOM representation. It uses pluggable source parsers for converting the DOM into the object representation.

- **Source Parser**

This pluggable component is responsible for parsing the input document (passed as an instance of `org.w3c.dom.Element`) into an object representation of the validation source. It is also responsible for resolving all references. The Source Loader instantiates the best source parser and passes control to it. For example, a WSDL source parser should resolve all `wsdl:import/include` elements.

## Tools, Tasks, and Reports

SOA Systinet enables you to monitor your SOA through the use of tools, tasks, and reports.

*Tools* provide additional functionality for the repository content. They keep the repository content up to date, run scheduled reporting, and execute customized tools. Also, impact management helps users understand the dependencies between artifacts and enables them to assess the impact of changes to their artifacts (for example, what happens if an XSD document is modified).

A *task* specifies the execution of a tool on selected artifacts or resources.

The result of executing a tool is a *report*.

Every use case is provided by a particular tool. Tools are organized as special artifact types, where each tool is represented by an individual artifact.

SOA Systinet includes the following tools, which are described in full in the *HP SOA Systinet User Guide*:

- **Impact Tools**

*Impact Management* tools enable you to perform impact and dependency analysis.

- **Sync Tools**

Change management tools are used to keep the blueprint of the corporate SOA being maintained by SOA Systinet in synchronization.

- **Job Tools**

Job Tools are a way of using customized Java classes to perform specific functions.

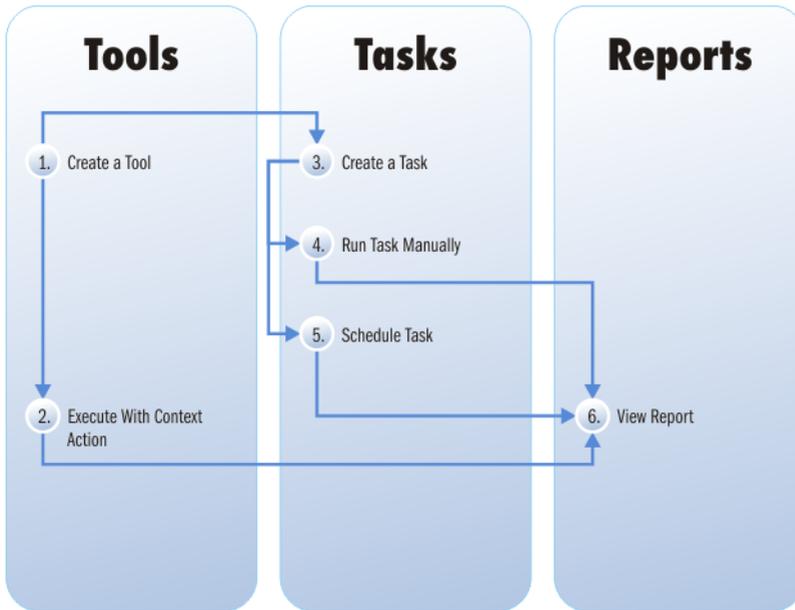
- **Reporting Tools**

Reporting Tools access scripts created on the reporting server.

There are several ways to make use of these tools in order to produce reports. The different ways to execute a tool to produce a report shown in [Figure 13, “Tool Execution Workflow”](#).

The procedure for each step is described in detail in the *HP SOA Systinet User Guide*.

**Figure 13. Tool Execution Workflow**



**Key to Figure 13, “Tool Execution Workflow”:**

- 1 You can create several types of tools or use the tools provided with SOA Systinet. The creation of each tool is explained in detail in the *HP SOA Systinet User Guide*.

- **Impact Tools**
- **Sync Tools**
- **Job Tools**
- **Reporting Tools**

2 Using a context action, it is possible to execute a tool from the detail view of artifacts. The tools you can run depend on the artifact. For details, see the description of each tool in the *HP SOA Systinet User Guide*.

Performing these actions immediately produces a report of the results of the execution of that tool on the artifact.

- 3 An alternative to running a tool directly is to create a task. This is more useful than directly running a tool because it is possible to create a particular task requiring a document or collection to be checked using the tool on a periodic basis. For details, see *Creating a Task* in the *HP SOA Systinet User Guide*.
- 4 After you create a task you can execute it at any time, immediately producing a report by clicking **Run** on the detail page of the task.
- 5 Optionally, set a time for the task to be executed or even establish a schedule for the task to be performed on a periodic basis. For details, see "Setting a Schedule" in the *HP SOA Systinet User Guide*.
- 6 The result of a tool or task execution is a report. For details, see "Reports" in the *HP SOA Systinet User Guide*.



These reports and the reports stored on the reporting server are not the same. The reports in the SOA Systinet UI refer to the progress and success of the execution of tools and tasks. The reports on the reporting server are specific instances of reports designed to be used in conjunction with the reporting tool.

## Versioning Overview

SOA Systinet maintains both versions and revisions. Whenever an artifact changes SOA Systinet creates a new revision. SOA Systinet also enables you to manually control independent versions of your SOA artifacts.

Two types of versioning are available, depending on the significance of the change to the artifact:

- **New Branch**

If you want to start development of a new version of a service, whilst keeping the existing version available, you can create a new branch.

Creating a new branch makes copies of the existing artifact, along with any related artifacts in the same governance process.

This enables you to start independent governance of the new version.

For example, version 1.0 of a policy artifact can describe a policy approved for production use, while version 2.0 of the same policy artifact can be in development.

- **Version Number Update**

Create a version number update when there is a significant change to a service that does not require new copies.

For example, change the version number of a service from 1.0 to 1.1 to reflect a change to a related endpoint.

For details, see "Managing Versions" in the *HP SOA Systinet User Guide*.



## 7 SOA Systinet Features

In addition to the features and concepts described in this guide, SOA Systinet also includes the following features:

- **Profile Management**

The Administrator can manage user and group profiles and configure access rights.

For details, see "Profile Management" in the *HP SOA Systinet Administrator Guide*.

- **Categorization**

Predefined taxonomies enable you to categorize and classify the artifacts in your SOA. This enables searching and filtering according to the needs of your organization. Taxonomies can be imported from the **Tools** tab.

For details, see "Categorizing Artifacts" in the *HP SOA Systinet User Guide*.

A separate client product, HP SOA Systinet Taxonomy Editor, is also available to create and manage taxonomies.

For details, see the *HP SOA Systinet Taxonomy Editor Guide*.

- **Registry Integration**

Customizable mapping between UDDI registries and SOA Systinet is used to enable imports and integration.

For details, see "Setting Up Registry Integration" in the *HP SOA Systinet Administrator Guide*.

- **HP Business Availability Center Integration**

SOA Systinet receives service availability statistics by integrating with BAC.

For details, see "Setting Up BAC/UCMDB Integration" in the *HP SOA Systinet Administrator Guide*.

- **HP Universal Configuration Management Database Integration**

SOA Systinet integrates with the UCMDB used by BAC for service discovery.

For details, see "Setting Up BAC/UCMDB Integration" in the *HP SOA Systinet Administrator Guide*.

- **HP SOA Policy Enforcer Integration**

SOA Systinet can access run-time policy enforcement information about services shared by HP SOA Policy Enforcer.

For details, see "Setting Up SOA Policy Enforcer Integration" in the *HP SOA Systinet Administrator Guide*.

- **HP Service Test Manager Integration**

SOA Systinet can register SOAP services in the Service Test Manager component of HP Quality Center for testing and receive the testing statistics.

For details, see "Setting Up STM Integration" in the *HP SOA Systinet Administrator Guide*.

- **LDAP Integration**

SOA Systinet can integrate with user accounts held on an LDAP server.

For details, see "LDAP Accounts Integration" in the *HP SOA Systinet Installation and Deployment Guide*.

- **API**

SOA Systinet provides a proprietary REST interface and an Atom-based REST Interface to access the repository. It is intended for developers who wish to manipulate and query the repository without using the web interface.

For details, see "REST Interface" in the *HP SOA Systinet Developer Guide*.

- **RSS Subscription and Notification**

RSS enables you to subscribe to constantly updated information giving notification of changes to repository data.

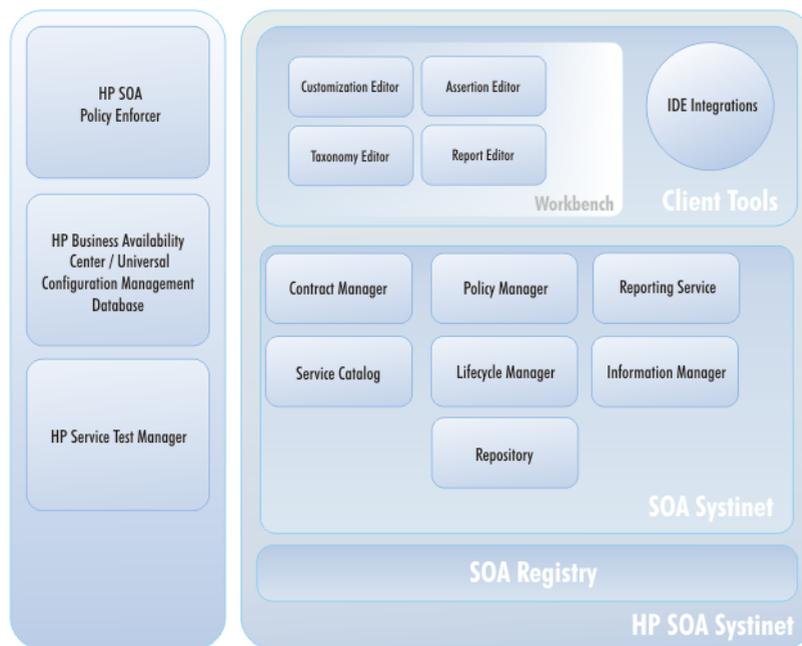
For details, see "RSS" in the *HP SOA Systinet Developer Guide*.



## 8 HP SOA Product Family

SOA Systinet is part of a larger family of SOA products designed to cover the entire range of SOA requirements.

**Figure 14. HP SOA Products**



### Standard Components

HP SOA Systinet is designed to enable SOA to be established and governed in a production environment with the following standard components:

- **Service Catalog**

Provides an entry point for the provision and management of services.

- **Information Manager**

Provides a suite of governance tools and low level access to the repository.

- **Reporting Service**

Monitors and reports on your SOA.

- **Repository**

Organizes and manages your SOA data.

- **Lifecycle Manager**

Enables you to control the service lifecycle and establish governance procedures for each lifecycle stage.

### **Optional Components**

In addition, your SOA Systinet license can include the following components:

- **Policy Manager**

Enables your production environment to conform to your business policy.

- **Contract Manager**

Enables the management of provider-consumer relationships.

### **Tools**

HP Software provide a set of tools for use with SOA Systinet:

- **HP SOA Systinet Workbench**

A suite of editor tools distributed as an Eclipse development platform or as a plugin for Eclipse.

Workbench includes the following components:

- **Customization Editor**

Enables you to customize the SOA model and the navigation and appearance of the user interface in SOA Systinet.

- **Assertion Editor**

Enables you to create the building blocks of technical and business policies.

- **Taxonomy Editor**

Enables you to create customized taxonomies to categorize your SOA artifacts according to your needs.

- **Report Editor**

Enables you to create customized reports. These can be deployed to the reporting service and then used in SOA Systinet.

- **IDE Integrations**

A set of plug-ins for different development environments offering SOA Systinet integration functionality.

HP Software provide the following IDE integration products:

- **HP SOA Systinet Plug-In for Eclipse**

Enables you to search the SOA Systinet repository, generate service clients and skeletons from SOA Systinet resources, perform local resource validation against SOA Systinet policies, and publish local resources to the SOA Systinet repository.

- **HP SOA Systinet Plug-In for Visual Studio**

Enables you to search the SOA Systinet repository, generate web references from SOA Systinet resources, and publish local resources to the SOA Systinet repository.

### **Integrated HP SOA Products**

SOA Systinet integrates with the following products:

- **HP SOA Registry Foundation**

Publish and index your services in a central location.

- **HP Business Availability Center and HP Universal Configuration Management Database**

Discover services stored in UCMDB and access service availability statistics generated by BAC to monitor actual performance against your service level objectives.

- **HP SOA Policy Enforcer**

Enforce your company policy on run-time services. The integration enables SOA Systinet to access the SOA PE information about shared services.

- **HP Service Test Management**

Register your SOAP services in STM for testing and validation to improve the quality and manage the complexity associated with SOA.