

HP SOA Systinet

Software Version: 3.00

Concepts Guide

Document Release Date: June 2008
Software Release Date: June 2008



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.

Third-Party Web Sites

HP provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. HP makes no representations or warranties whatsoever as to site content or availability.

Copyright Notices

© Copyright 2003-2008 Hewlett-Packard Development Company, L.P.

Trademark Notices

Java™ is a US trademark of Sun Microsystems, Inc. Microsoft®, Windows® and Windows XP® are U.S. registered trademarks of Microsoft Corporation. IBM®, AIX® and WebSphere® are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries. BEA® and WebLogic® are registered trademarks of BEA Systems, Inc.

Contents

About This Guide.	5
How This Guide is Organized.	5
Document Conventions.	7
Documentation Updates.	8
Support.	9
1 SOA and SOA Systinet.	11
2 User Perspectives.	13
3 SOA Definition Model.	15
Artifacts.	15
Properties.	16
Taxonomies.	16
4 SOA Repository.	19
5 Services.	21
Service Implementation.	22
Service Discovery.	24
Contracts.	25
6 Governance.	29
Lifecycle Overview.	29
Policy Overview.	32
Tools, Tasks, and Reports.	39
Versioning Overview.	41
7 Additional Features.	43

8 HP SOA Systinet Product Family..... 47

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.

How This Guide is Organized

The HP SOA Systinet Concepts Guide gives an overview and introduction to the concepts used by SOA Systinet to manage an SOA instance.

This guide includes:

- [Chapter 1, SOA and SOA Systinet](#)
An introduction to SOA Systinet.
- [Chapter 2, User Perspectives](#)
The use of user specific perspectives in SOA Systinet.
- [Chapter 3, SOA Definition Model](#)
An introduction to the SOA Definition Model used in SOA Systinet.
- [Chapter 4, SOA Repository](#)
The content management system for SOA.
- [Chapter 5, Services](#)

An overview of how SOA Systinet manages services and contracts.

- [Chapter 6, Governance](#)

An introduction to the governance of artifacts in SOA Systinet.

- [Chapter 7, Additional Features](#)

An introduction to additional features of SOA Systinet.

- [Chapter 8, HP SOA Systinet Product Family](#).

An overview of the HP family of SOA products.

Document Conventions

This document uses the following typographical conventions:

run.bat make	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>rmdir /S /Q System32</code>	User input.
<code>C:\System.ini</code>	Filenames, 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.
Shift+N	Combination of keystrokes.
Service View	Label, word, or phrase in a GUI window, often clickable.
OK	Button in a user interface.
New→Service	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 login page.

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

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

To register for an HP Passport ID, go to:

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

1 SOA and SOA Systinet

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 becomes the place where service providers and consumers obtain a common view of *business services* they offer and consume.

- **Discovery**

SOA Systinet indexes published artifacts so they can be browsed and searched using both full text search and query languages.

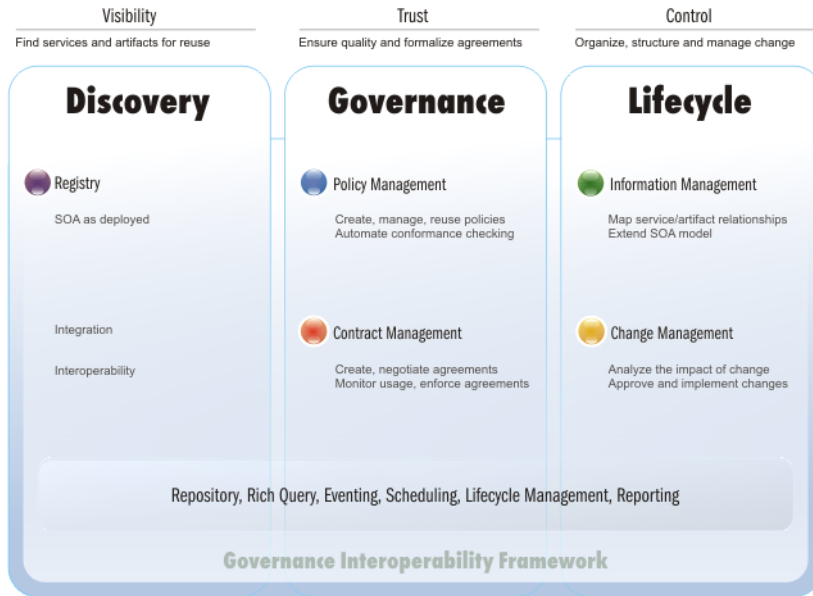
- **Full history**

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 becomes self-describing. In the case of service artifacts, this enables service consumers to obtain 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. Despite the fact that the number of services grows and their relationships and dependencies are more and more complex, the growth of this production within SOA must be controlled as a whole. 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—it is a governance platform used to manage vital SOA.

2 User Perspectives

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.

To meet the needs of typical users, a number of *perspectives* are defined by default. These perspectives are designed to meet the needs of typical user roles. This documentation also uses these perspectives to explain SOA processes, but like other aspects of the system, perspectives are designed to be customizable.

Furthermore, a user is not forced to assume a particular perspective. The UI enables information to be viewed from any perspective based on permissions granted by the Administrator. A user can switch to any permitted perspective at any time. 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 default perspectives offered by SOA Systinet are:

- **Administrator**

The Administrator perspective is designed for the administration of users, groups, and perspectives in SOA Systinet as well as other critical functions. The 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 and not necessarily in 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 to the general user.

- **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.

The Administrator can manage the membership of perspectives in SOA Systinet.

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

You can configure existing perspectives and create additional ones using HP SOA Systinet Customization Editor.

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

3 SOA Definition Model

At the core of 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. Hence 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 15](#)
- [Properties on page 16](#)
- [Taxonomies on page 16](#)

Artifacts

SOA Systinet enables your SOA to be described using the following aspects:

- **Metadata**

The attributes that describe the artifact and its relationships to other artifacts.

- **Abstract**

This type of artifact does not have actual instances in the repository. They are used as parent types for other artifact types that inherit their properties.

- **Instances**

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

The artifact types form a hierarchical taxonomy, described in detail in "SDM 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 which 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 enabling, for example, 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 SOA, reflecting the SOA Definition Model. Hence 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 allowing you to access the data you need quickly and efficiently.

The repository includes the following resource types:

- **Documents**
- **Collections** – containing documents or other collections

A *document* is like a file, and analogous to a row in a relational database. Each document can include SDM metadata and relationships with other documents.

A *collection* is like a directory. They are analogous to tables in a relational database, but they can contain other collections and hence 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

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.

- **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 22](#)

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

- [Service Discovery on page 24](#)

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

- [Contracts on page 25](#)

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

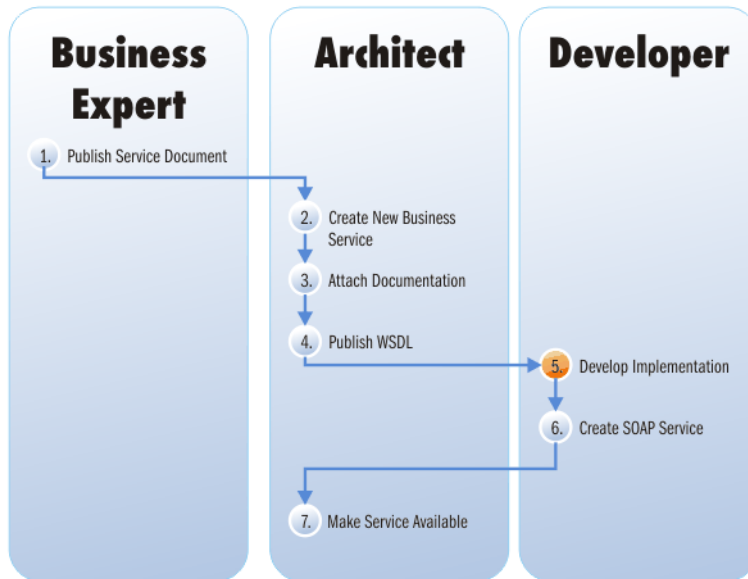
Service Implementation

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.

The workflow diagram 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 2. Business Service Workflow



- **Business Expert**

- 1 The business expert publishes a service document in SOA Systinet describing a proposed new service.

- **Architect**

- 2 The architect receives notification of the new service requirement and having reviewed the documentation creates a new business service.
- 3 The existing service document and a new technical document is created and attached to the service.
- 4 A WSDL document is published containing a technical description of the service.

- **Developer**

- 5 The developer then reviews the documentation and uses the WSDL to develop an implementation of the business service (for example, in MS Visual Studio).
 - 6 After completing development, a SOAP service is added to the business service with relevant documentation.
- **Architect**
 - 7 The business service is then made available by editing it and changing its Lifecycle Stage to production and setting the Enable Consumption Requests flag.

Service Discovery

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

The following methods are available:

- **Uploading Service Definition Documents**

SOA Systinet includes a publisher which 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 22](#) describes the process of populating SOA Systinet with services. A key service relationship is the provider-consumer relationship that can be created 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, and 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

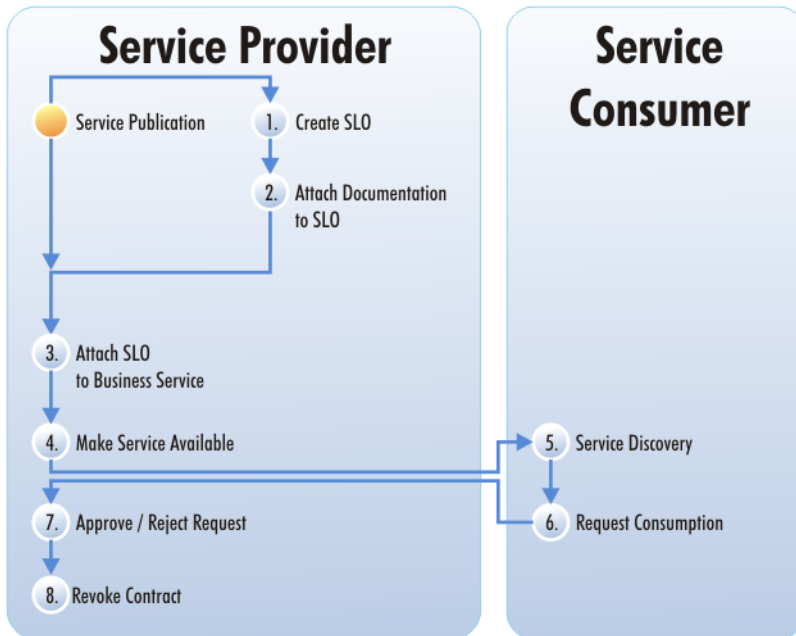
can establish a contract. The [Figure 3](#) displays the typical tasks that should be performed, and an overview of each step is given below.

- ▶ 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 3. Contract Workflow



- **Service Provider**

- 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. An SLO can be created independently, or as part of a service.
 - 2 The SLO can also have documentation attached to it.
 - 3 The SLO is attached to the Business Service.
 - 4 The Business Service is then made available by making it ready for consumption.
- **Service Consumer**
 - 5 The consumer first 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*.
 - **Service Provider**
 - 7 When a consumer requests consumption of a service, the provider must either approve or reject the request. See *Processing Consumption Requests*.
 - 8 The provider also can also cancel an existing contract. See *Revoking an Active Contract*.

6 Governance

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

This chapter introduces the key concepts in SOA Systinet governance:

- [Lifecycle Overview on page 29](#)

An introduction to lifecycle governance in SOA Systinet.

- [Policy Overview on page 32](#)

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

- [Tools, Tasks, and Reports on page 39](#)

The SOA Systinet suite of tools allows 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 41](#)

An introduction to version management in SOA Systinet.

Lifecycle Overview

The process of populating SOA Systinet with service infrastructure is described in [Service Implementation on page 22](#).

The development of service artifacts can be further controlled in SOA Systinet using Lifecycle Governance.

The lifecycle is a set of stages and promotion rules for each stage of selected artifact types. Rules of promotion are defined using policies, tasks, and approvers and can be associated with any stage of any artifact type involved in the governance process.

The governance process itself has a lifecycle, which is broken down to the following key stages:

- **Draft**

The creation of a governance process can be complex and cannot be created in one step.

Whilst in this stage, the process cannot be used for governance but it does enable you to make unrestricted changes to the process.

- **Published**

All governance processes in this stage can be used for the governance of artifacts. However, you are unable to make major modifications to the process whilst it is published. For example, you cannot add or remove lifecycle stages. To do this, you should use the Deprecate and Clone functionality.

- **Deprecated**

This is the final stage of the lifecycle of a governance process. When a process enters this stage, it reverts to draft status, and can no longer be used in the governance of artifacts, unless it is published again.

These stages are key in the overall lifecycle management process, which is broken down to the following steps:

- 1 The first step in lifecycle management is to define a governance process. This process defines the lifecycle stages that make up the process, with each stage further defining the approvers and any associated policies and tasks.

For more details on defining a governance process, see *Creating a Governance Process in the HP SOA Systinet User Guide* .

- 2 The next step in the process is publication. In short, publication enables the use of the process in the governance of artifacts.

For more details on publishing a governance process, see *Publishing a Governance Process in the HP SOA Systinet User Guide* .

For more details on governing artifacts, see *Governing Artifacts in the HP SOA Systinet User Guide*.

- 3 After a lifecycle process instance is in use, you can promote service artifacts to the next lifecycle stage.

Part of this process is to submit a promotion request to approvers. If all associated policies and tasks have been completed, the approvers can vote for the artifact to be promoted to the next stage.

For more details on submitting a promotion request, see *Submitting a Promotion Request in the HP SOA Systinet User Guide*.

For more details on policies and tasks, see *Managing Associated Policies and Tasks in the HP SOA Systinet User Guide*.

For more details on approving a promotion request, see *Handling Promotion Requests in the HP SOA Systinet User Guide*.

- 4 If you need to modify a governance process, you can use the deprecation function.

For more details on deprecation, see *Deprecating a Governance Process in the HP SOA Systinet User Guide*.

SOA Systinet uses specific terminology in lifecycle governance. A brief description is given below:

- **Lifecycle Process Instance**

An instance created when the process is used for the governance of a specific group of artifacts.

- **Governance Tree**

A grouping which enables the group of artifacts to be used in other tasks. Each governance tree has a root artifact - usually the highest level abstraction in the Service level, for example, Business Service - which acts as the point from which promotion is initiated.

- **Stage**

A milestone in the development of each artifact. A stage is marked as complete when the validation of associated policies and completion of associated tasks has been confirmed by approvers.

- **Promotion**

A phase in the lifecycle process, initiated by the 'Start Promotion' action. All associated policies are automatically scheduled for validation and approvers are notified of the promotion request. When promotion is successfully completed, all artifacts in the governance tree move to the next stage.

- **Approvers**

Specified users - individuals and groups - that decide whether to approve or reject a promotion request.

- **Task**

A unit of work to be completed by a specified user. Although completion tracking can be set in the lifecycle stage definition, the task must be manually marked as complete by the artifact maintainer before promotion can be started.

- **(Technical) Policy Associations**

Similar to business policies but can be set for a specific artifact type or stage. An associated policy can be set as 'required', meaning that if the artifact does not pass validation using the specified policy, promotion will fail.

- **Artifact Maintainer**

A user with write permission on an artifact.

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 33.](#)

When and why you use Policy Manager

- [Policy Concepts on page 33](#)

The concepts Policy Manager uses.

- [Policy Workflow on page 35](#)

The lifecycle of policy development and report generation.

- [Policy Manager Components on page 37](#)

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 33](#)
- [Business and Technical Policies on page 34](#)
- [Compliance Validation and Reports on page 34](#)

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 allow 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 manually create your own validation handlers, 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:

- **The identification of a group of artifacts**
- **Sets of requirements that apply to the identified artifacts**

In the Policy Manager model, the identification of artifacts (1) is called a *Scope of Application* and the sets of requirements (2) 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 (1) all WSDLs must have documentation and (2) 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.

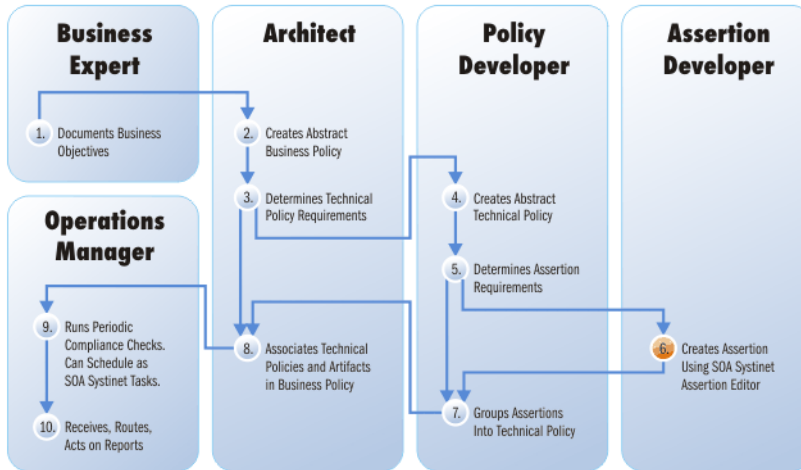
In the course of policy development, you might want to test an artifact by checking it against one or more technical policies. This kind of compliance check is also possible in Policy Manager. Compliance validation can be performed automatically or manually, depending on the type of assertion involved.

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

Policy Workflow

The workflow of policy creation and its subsequent use is shown in [Figure 4](#).

Figure 4. Policy Workflow



The policy workflow is broken down to the following steps:

- 1 The business expert documents business objectives in the SOA Systinet repository. They then instruct 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, they can move on to Step 8. Otherwise, they instruct 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, they can move on to Step 7. Otherwise, they instruct 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 UI for browsing, viewing, and editing business and technical policies. It can be accessed from a menu of links in all the Policy Manager and part of the SOA Systinet UI.

The policy editor does the following:

- Uploads the policy to the UI session.
- Allows the policy to be edited in the session.
- Saves the policy in the SOA Systinet repository.
- Allows 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 allows 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**

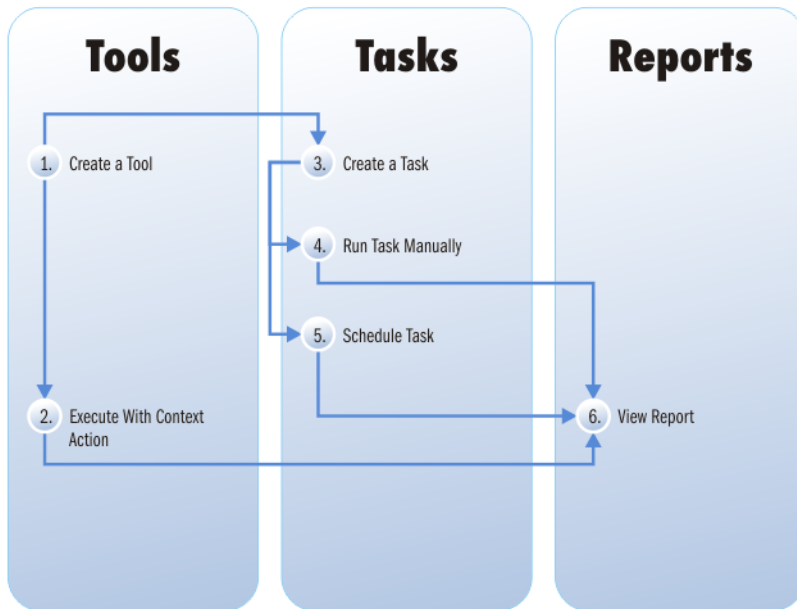
Job Tools are a way of using customized Java Classes to perform specific functions.

- **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 the [Figure 5](#) diagram below:

Figure 5. Tool Execution Workflow



Each step is described in detail in the *HP SOA Systinet User Guide* .

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 Alternatively, 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 Additional Features

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

- **Profile Management**

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

For details, see "Profile Management" in the *HP SOA Systinet Administration 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**

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**

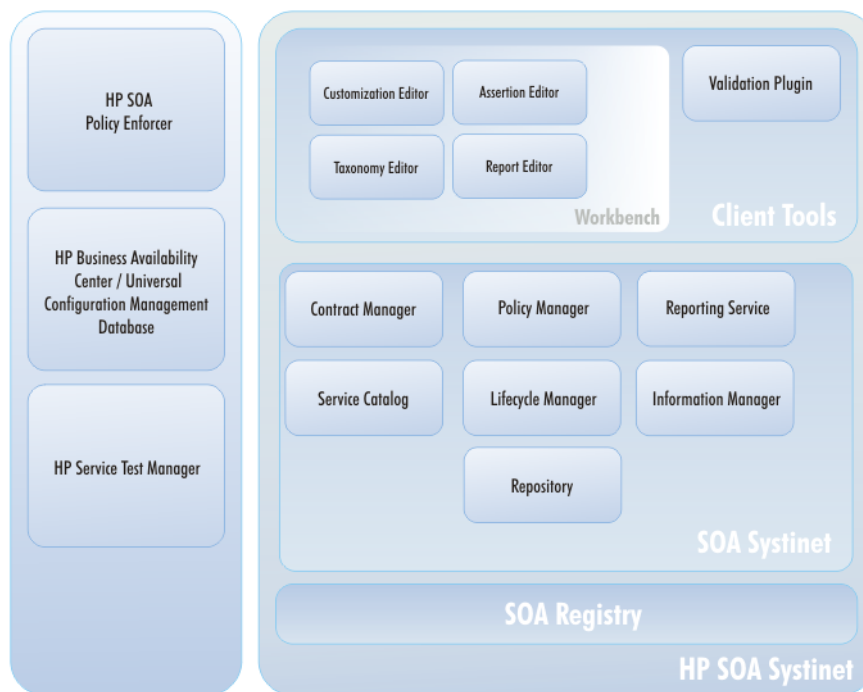
RSS enables you to subscribe to constantly updated information giving notification of changes to repository data.

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

8 HP SOA Systinet Product Family

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

Figure 6. HP SOA Systinet Products



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

- The **Service Catalog** is the entry point for the provision and management of services.
- **Information Manager** provides a suite of governance tools and low level access to the repository.
- The **Reporting Service** monitors and reports on your SOA.
- The **Repository** organizes and manages your SOA data.
- **Lifecycle Manager** enables you to control the service lifecycle and establish governance procedures for each lifecycle stage.

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.

Included with SOA Systinet are a set of **Client Tools** enabling you to customize your SOA environment:

- **HP SOA Systinet Workbench** is a suite of editor tools distributed as an Eclipse development platform or as a plugin for Eclipse. It includes the following:
 - **Customization Editor** enables you to customize the SOA model and the navigation and appearance of the user interface in SOA Systinet.
 - **Assertion Editor** enables the creation of the building blocks of technical and business policies.
 - **Taxonomy Editor** enables the creation of customized taxonomies to categorize your SOA artifacts according to your needs.
 - **Report Editor** enables the creation of customized reports. These can be deployed to the reporting service and then used in SOA Systinet.
- **Validation Plugin** enables policy validation to be run locally without needing to start SOA Systinet. This enables documents to be validated against business policies before they are added to the repository.

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.