# HP SOA Systinet

Software Version: 2.52, Standard Edition

## Concepts Guide

## 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*

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

*Copyright Notices*

Copyright © 2006-2007, 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

# Welcome to 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.

## How This Guide is Organized

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

This guide includes:

# Document Conventions

The typographic conventions used in this document are:

| run.bat make | Script name or other executable command plus mandatory arguments. |
|---|---|
| [--help] | A command-line option. |
| either \| or | A choice of arguments. |
| *replace_value* | A command-line argument that should be replaced with an actual value. |
| {arg1 \| arg2} | A choice between two command-line arguments where one or the other is mandatory. |
| `rmdir /S /Q System32` | Operating system commands and other user input that you can type on the command line and press **Enter** to invoke. Items in *italics* should be replaced by actual values. |
| `C:\System.ini` | Filenames, directory names, paths and package names. |
| `a.append(b);` | Program source code. |
| `server.Version` | An inline Java or C++ class name. |
| `getVersion()` | An inline Java method name. |
| **Shift**-**N** | A combination of keystrokes. |
| **Service View** | A label, word or phrase in a GUI window, often clickable. |
| **OK** | A button in a GUI window. |
| **New**->**Service** | Menu choice. |

# Documentation Updates

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

• Software version number

• 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://ovweb.external.hp.com/lpe/doc_serv/**

# Support

## Mercury Product Support

You can obtain support information for products formerly produced by Mercury as follows:

- If you work with an HP Software Services Integrator (SVI) partner (**http://h20230.www2.hp.-com/svi_partner_list.jsp**), contact your SVI agent.

- If you have an active HP Software support contract, visit the HP Software Support Web site and use the Self-Solve Knowledge Search to find answers to technical questions.

- For the latest information about support processes and tools available for products formerly produced by Mercury, we encourage you to visit the Mercury Customer Support Web site at: **http://hp.-com/go/hpsoftwaresupport**.

- For the latest information about support processes and tools available for products formerly produced by Systinet, we encourage you to visit the Systinet Online Support Web site at: **http://www.systinet.-com/support/index**.

- If you have additional questions, contact your HP Sales Representative.

## HP Software Support

You can visit the HP Software Support Web site at:

**http://www.hp.com/go/hpsoftwaresupport**

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support 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**
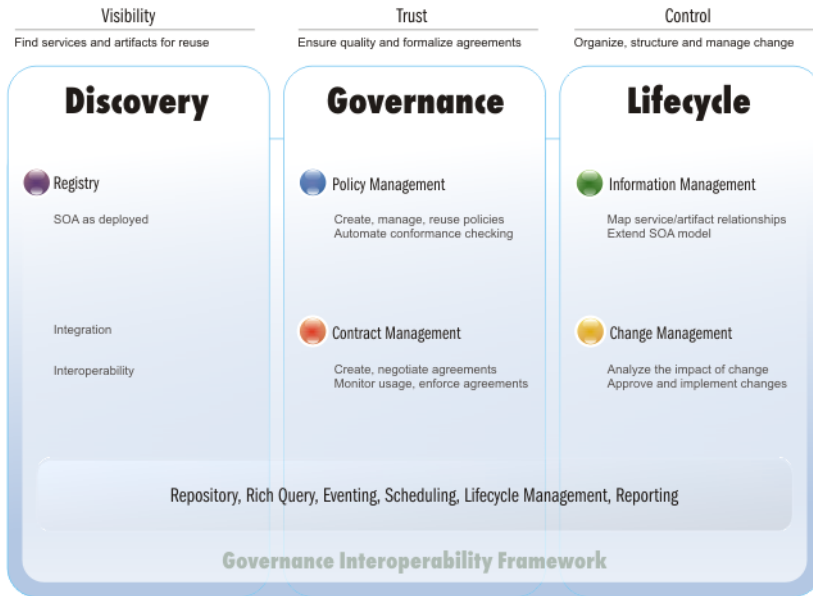


After SOA infrastructure becomes an important part of enterprise internal systems, its functionality is critical. The blueprint of SOA artifacts (providing visibility, discovery and reuse) is no longer sufficient. Despite the fact that the number of services grows and their relationships and dependencies are more and more complex, the growth of this production 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 perspectives is designed for the administrator of the 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.

  > **Note:** 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.

- **Guest**. The guest perspective enables access to certain pages for users who are not signed-in.

The administrator can configure perspectives in SOA Systinet. For more information, see the HP SOA Systinet Administration 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.

This section introduces the basic concepts of the SDM:

- Artifacts on page 15

- Properties on page 16

- Taxonomies on page 16

## Artifacts

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

- Metadata – these are 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, which is described in detail in SDM Model in the HP SOA Systinet Reference Guide. Each leaf 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. There are three kinds of property:

- **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.

# 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 managment, reliability, and failover.

Complimentary 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.

More details are given in the HP SOA Systinet Reference Guide, which describes the structure of the repository and the format of the resources it contains.

# 5 Services

A business service in SOA Systinet is a high-level abstraction of a service in business rather than technical terms.

SOA Systinet includes the Service Catalog to publish and implement your services and then make them available to consumers, as follows:

- **Service Implementation.** The service catalog is the gateway to service provision. From this view of services it is possible to manage all the associated implementations, documentation, and usage plans.

- **Service Lifecycle.** SOA Systinet enables you to manage your service development by creating approval processes which can then be associated with business services.

- **Contract Management.** SOA Systinet enables you to establish contracts with associated usage plans and service level agreements.

This chapter introduces these concepts in the following sections:

- A brief look at services in SOA Systinet.

- An introduction to lifecycle management in SOA Systinet.

- An introduction to contract management in SOA Systinet.

## 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 services to be added to your SOA and made available to consumers. The workflow diagram describes a typical publication procedure and the steps required to achieve this. 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**
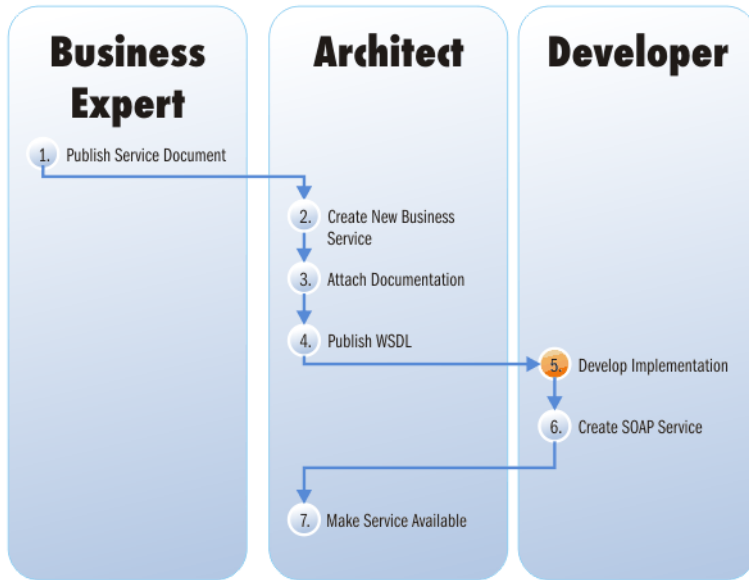


- Business Expert

    1    The business expert publishes a service document in SOA Systinet describing a proposed new service. See Adding Documentation.


- Architect

    2    The architect receives notification of the new service requirement and having reviewed the documentation creates a new business service. See Creating a Business Service.

    3    The existing service document and a new technical document is created and attached to the service. See Adding Service Documentation.

    4    A WSDL document is published containing a technical description of the service. See Publishing Definition Resources.

- 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   Having completed development a SOAP service is added to the business service with relevant documentation. See Adding a SOAP Service.

- 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 (Standard Edition only). See Making the Service Available.

## Service Lifecycle and Approvals

Service Implementation on page 19 describes the process of populating SOA Systinet with services. The development of services can be further controlled in SOA Systinet using the lifecycle and approvals process included with the product.

The lifecycle functions enable you to create a set of approval processes with defined users set as approvers for each lifecycle stage in the process. Business services can then be associated with a particular approval process.

The lifecycle diagram displays the workflow of typical tasks that should be performed and an overview of each step is given below. The procedure for each step is described in detail in the HP SOA Systinet User Guide.

**Figure 3. Lifecycle Workflow**
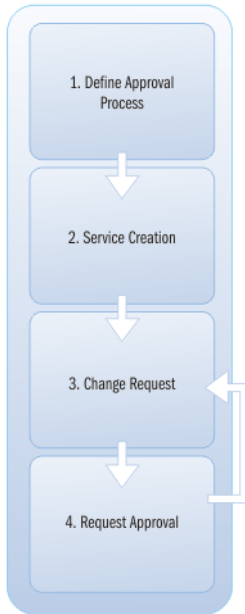


1    The first step in lifecycle management is to define an approval process. This process defines which lifecycle stages require approval before further changes can be made to it. See Creating an Approval Process Definition.

2    During service publication the service can be associated with a particular approval process. See Creating a Business Service.

3    When a service is at a lifecycle stage that requires approval, a user can make a change and should then request approval for that change. See Submitting a Change Request. E-mails are sent to the approvers and while the approval is pending no further changes can be made to the service.

4    The approver reviews the change request, accessing any reports, policy information, or notes by other approvers as necessary to inform their decision, and then either approves or denies the request. See Handling Change Requests.

# Contracts

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 drivers that motivate publishers 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 a consumer access to a service . Helps 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. The reports enable you to charge back based on the monitoring of outputs.

- **Consumption Auditing.** Providers need to know who is using specific services and who granted them access. They need to know for many reasons, such as security, accounting, capacity planning and change management.

SOA Systinet provides Contract Manager to facilitate these functions.

Contract Manager enables providers to specify conditions under which the service can be consumed as service usage plans. Consumers can request the consumption of a particular service by choosing one of the usage plans provided and the consumer establishes a *contract* describing what is provided and under what conditions the service can be consumed.

For a 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 workflow diagram displays the typical tasks that should be performed, and an overview of each step is given below. The procedure for each step is described in detail in the HP SOA Systinet User Guide.

**Figure 4. Contract Workflow**
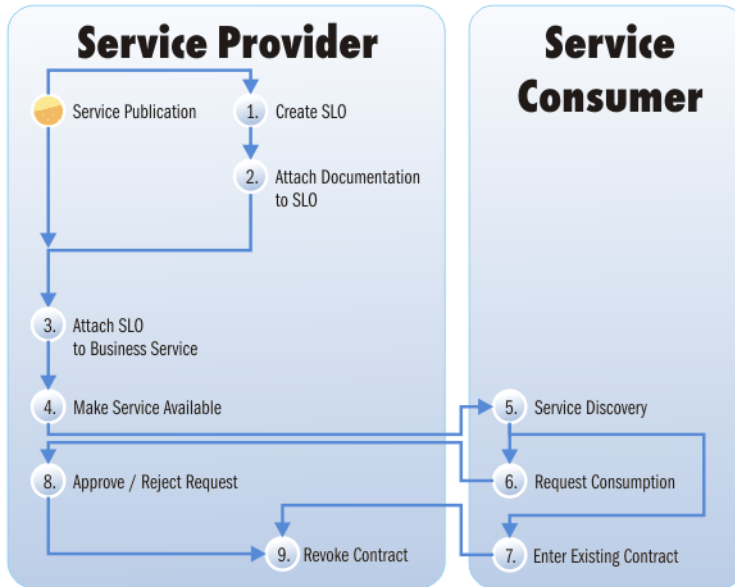


- Service Provider

    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 as part of a service, as described in Adding an SLO, or independently, as described in Creating an Artifact.

    2  The SLO can also have documentation associated with it. See Attaching Documentation to Artifacts.

    3  The SLO is attached to the Business Service. See Adding an SLO.

    4  The Business Service is then made available by making it ready for consumption. See Making the Service Available.

- Service Consumer

5   The consumer first browses SOA Systinet to see which services are available and whether they meet their requirements. See Full Text Search.

6   After the consumer finds the service, they can request it. See Requesting Consumption.

7   Alternatively, the consumer can look at existing contracts for the service and select one instead of making a request. See Importing a Contract.

• Service Provider

8   When a consumer requests consumption of a service, the provider must either approve or reject the request. See Handling a Request.

9   The provider also can also cancel an existing contract. See Revoking a 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:

- **Policy Overview on page 27.** 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 33.** 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.

## 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 27. When and why you use Policy Manager.

- Policy Concepts on page 28. The concepts Policy Manager uses.

- Policy Lifecycle on page 30. The lifecycle of policy development and report generation.

- Policy Manager Components on page 31. 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 time.

- **To support and maintain the quality of SOA Systinet Platform**. Policy Manager can check the contents of the SOA Systinet Platform repository for their compliance with publishing policies.

## Policy Concepts

HP SOA Systinet Policy Manager uses the following concepts in its model, from low-level to high:

### Assertions

The individual requirements that comprise a technical policy are called *assertions*. Assertions contain definitions of how they 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 Management 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.

Users can also manually create their own validation handlers, which can be added as expansions in HP SOA Systinet Assertion Editor.

## Business and Technical Policies

HP SOA Systinet 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.**  The payoff for setting up a business policy. Artifacts identified in a business policy are checked for compliance with the individual assertions in the associated technical policies. In enterprise terms, this is where we find out if the business objectives set forth in policies are being met.

- **Resource compliance.** This is 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. Operationally, 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, the user 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 computationally or manually (decided by a user), depending on the type of assertion involved.

## Policy Lifecycle

The lifecycle of policy creation and use is shown in Figure 5.

**Figure 5. Policy Lifecycle**
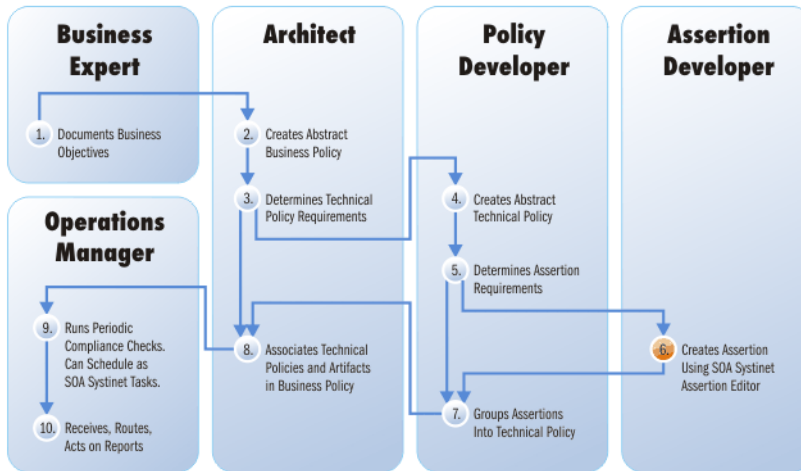


The steps of the policy lifecycle are:

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 name and description, related to one or more business objectives.

3   The architect now determines what technical policies are needed for his business policy. If these technical policies already exist in the repository, they can move on to 8. Otherwise, they instruct the policy developer to create them.

4   The policy developer creates an abstract technical policy, with only 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 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.

7    The policy developer groups assertions into technical policies. They run 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. They run compliance checks to test the business policy before publishing it.

9    The operations manager can now use the business policy to enforce business objectives. They 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 Editor/Viewer.** This is a UI for browsing/viewing/editing business and technical policies. It can be called from a menu of links in all the Policy Manager and part of the SOA Systinet Platform 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, a client 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 human 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. The tools can 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; 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.

The automation of the governance tasks to ensure that the corporate SOA is under continuous control, as described in Scheduling a Task in the HP SOA Systinet User Guide.

There are several ways to make use of these tools in order to produce reports. The workflow diagram shows the different ways to execute a tool to produce a report. Each step is described in detail in the SOA Systinet User Guide.

**Figure 6. Tool Execution Workflow**
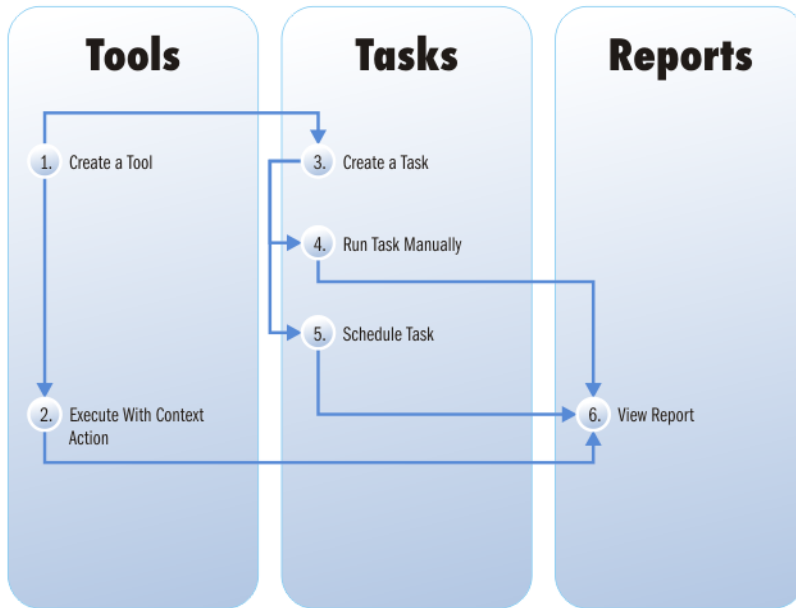


1  You can create several types of tools or use the tools provided with SOA Systinet. The creation of each tool is explained in detail in the HP Systinet User Guide:

   • 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. 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 that 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. See Creating a Task.

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. See Setting a Schedule.

6   The result of a tool or task execution is a report. See Reports.

> ➤   **Note:** 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.

# 7 Additional Features

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

- **User Management**. The administrator can manage users and groups and configure registration characteristics and access rights. See User Account Management in the HP SOA Systinet Administration Guide for details.

- **Single Sign-On**. SOA Systinet uses SSO to enable you to access different applications using a single login.

- **Categorization**. Predefined taxonomies enable you to categorize and classify the artifacts in your SOA. This enables lifecycle management, and searching and filtering according to the needs of your organization. Taxonomies can be imported from the **Tools** tab. A separate client product, HP SOA Systinet Taxonomy Editor, is also available to create and manage taxonomies.

- **Registry Integration**. Customizable mapping between UDDI registries and SOA Systinet is used to enable imports and integration. The Registry Integration chapter in the HP SOA Systinet User Guide describes this feature in detail.

- **HP Business Availability Center Integration**. SOA Systinet utilizes service availability statistics by integrating with HP BAC. The BAC Integration chapter in the HP SOA Systinet User Guide describes this feature in detail.

- **HP SOA Manager**. SOA Systinet can access information about services shared by HP SOA Manager. The SOA Manager Integration chapter in the HP SOA Systinet User Guide describes this feature in detail.

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

- **API**. The REST HTTP Interface provides another way to access the repository. It is intended for developers who wish to manipulate and query the repository without using the web interface. The REST Interface chapter in the HP SOA Systinet Developer Guide provides detailed information about the features and uses of the API.

- **RSS Subscription and Notification**. RSS enables you to subscribe to constantly updated information giving notification of changes to repository data. The RSS chapter in the HP SOA Systinet Developer Guide describes the types of feeds supported and how to use them.

# 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 7. HP SOA Systinet Products**
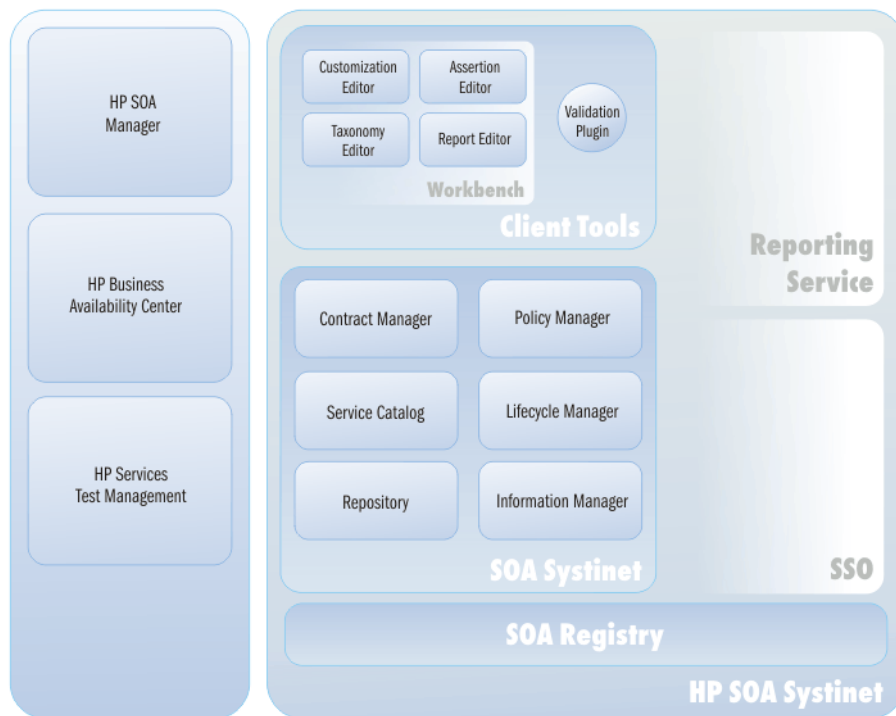


**HP SOA Systinet**

Designed to enable SOA to be established and governed in a production environment with the following components:

- The **Visibility Edition** includes:

  - 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 report on your SOA.

  - The **SSO Service** manages access to Systinet products with a single log-on per user.

  - The **Repository** organizes and manages your SOA data.

- The **Standard Edition** includes the following additional components:

  - **Policy Manager** enables your production environment to conform to your business policy.

  - **Contract Manager** enables the management of provider-consumer relationships.

  - **Lifecycle Manager** enables you to control the service lifecycle and establish approval procedures for each lifecycle stage.

  - **HP SOA Manager Integration** enables you access information on shared services in HP SOA Manager.

**Client Tools**

Included with SOA Systinet are a set of client tools enabling you to customize your SOA environment:

- SOA Systinet Workbench is a suite of editor tools distributed as an Eclipse development platform. Each editor can also be installed to Eclipse as an individual plugin.

  - The **Customization Editor** enables you to customize the SOA model and the navigation and appearance of the user interface in SOA Systinet.

  - The **Assertion Editor** enables the creation of the building blocks of technical and business policies.

- The **Taxonomy Editor** enables the creation of customized taxonomies to categorize your SOA artifacts according to your needs.

- The **Report Editor** enables the creation of customized reports. These can be deployed to the reporting service and then used in SOA Systinet.

- The **Validation Plugin** enables policy validation to be run locally without needing to start SOA Systinet. This enables documents to be vaildated against business policies before they are added to the repository.

**HP SOA Systinet Registry** enables the publication and indexing of your services in a central location.

**HP Business Availability Center**. SOA Systinet integrates with HP BAC in order to access service availability statistics allowing you to monitor actual performance against your service level objectives.

**HP SOA Manager** enables you to manage and optimize your SOA resources. The integration enables SOA Systinet to access the SOA Manager information about services with the same name.

**HP Service Test Management** provides SOAP services testing and validation, enabling you to improve the quality and manage the complexity associated with SOA. The integration is performed from Service Test Management.

# Index

## A
assertion
    architecture, 27

## B
business policy
    architecture, 27
business service
    scenario, 19
    use case, 19

## C
compliance, compliance check (see validation)
contract
    scenario, 23
    use case, 23

## G
governance
    tasks, 33
    tools, 33

## P
perspective, 13

## R
report
    architecture, 27

## S
scenario
    business service, 19
    contract, 23
source parser
    architecture, 27

## T
technical policy
    architecture, 27
tool
    governance, 33

## U
use case
    business service, 19
    contract, 23
user perspective (see perspective)

## V
validation
    architecture, 27
validation handler
    architecture, 27