



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Conversation Rules Templates Guide

Conversation Rules—Overview of Genesys Elements

# Conversation Rules—Overview of Genesys Elements

## Contents

- **1 Conversation Rules—Overview of Genesys Elements**
  - 1.1 Genesys Composer
  - 1.2 Orchestration Server
  - 1.3 GVP
  - 1.4 SCXML applications
  - 1.5 Conversation Manager/UCS
  - 1.6 Genesys Rules
  - 1.7 Genesys Reporting

### Genesys Composer

Composer provides a set of function blocks that allow access to Context Services. These out-of-the box function blocks on the workflow diagram palette allow the developer to create applications that perform various actions, such as:

- Identify customers and update their profiles.
- Extend customer profiles with user-defined information.
- Query a customer's profile.
- Create, start, complete, and query customer services.
- Query customers' active services.
- Enter, complete, and query service states and specific tasks.
- Use a business rule block to request evaluation of business logic developed in Genesys Rules System by business analysts, and act on the result.

### Orchestration Server

Execute the orchestration application. Orchestration Server has a function in Conversation Manager similar to the function of Universal Routing Server (URS) in Genesys voice and multimedia solutions. One of the main differences is that it operates based on business processes developed in State Chart XML (SCXML) rather than routing strategies written in IRL (Intelligent Routing Language, a Genesys proprietary language).

### GVP

Executes the VoiceXML applications.

### SCXML applications

SCXML applications can be written directly using any XML or plain text editor, or with Genesys Composer, an Eclipse-based development environment. They are published on an application server such as JBoss or another Java-based application server, and are executed on Orchestration Server.

### Conversation Manager/UCS

Genesys Conversation Manager takes Genesys' core capability of routing and extends it, generalizes

---

it, and integrates it more tightly with other Genesys products. Rather than the call (T-Server) or the interaction (eServices/Multimedia), Conversation Manager takes the service as the basic entity. It orchestrates the service process across channels and over time, using dynamic data and business rules to make decisions about operations. For example;

A bank customer calls a toll-free number inquiring about mortgage preapproval. An IVR prompts him to enter his account number, then transfers him to an agent, who fills in an application form for him and asks him to fax some supporting documents. After he faxes the documents, he receives an SMS message thanking him and informing him that he will receive a response within 48 hours. The next day he receives an e-mail congratulating him on the approval of his application.

This example involves voice, IVR, fax, SMS, and e-mail channels. Conversation Manager is able to treat the entire sequence as a single service.

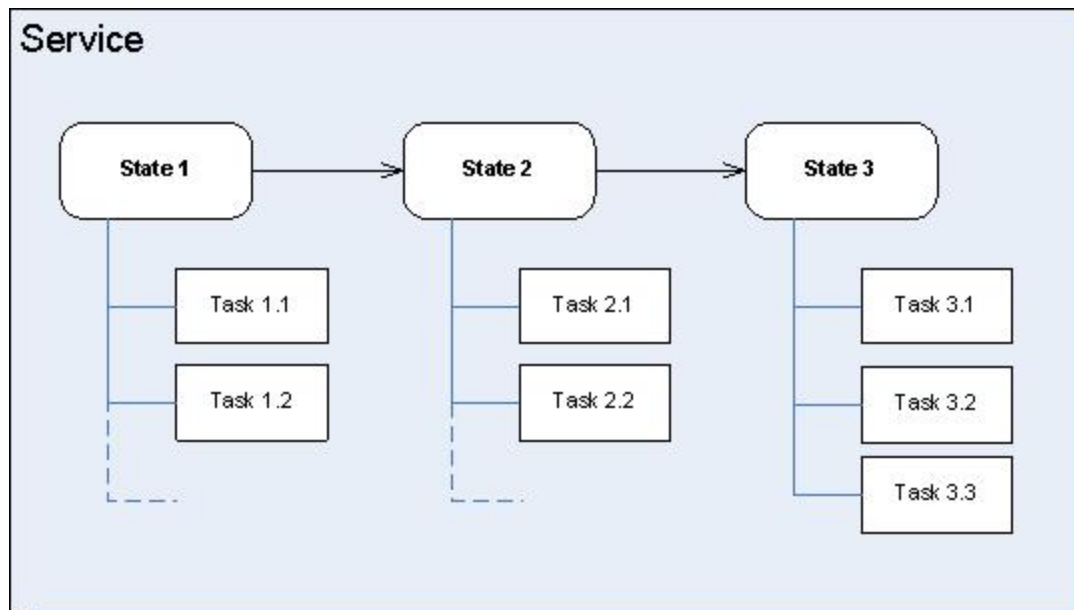
### Service/State/Task Model

Conversation Manager adds to Genesys the concept of service, which may be defined as follows:

- It represents a business process, which in turn may be seen as a communication or series of communications between a customer and an enterprise, and possibly also between various parts of the enterprise.
- It can span multiple interactions.
- It may include interactions in various media.
- It has a temporal beginning and end.
- It may be subdivided into states, which in turn may be subdivided into tasks.

Services are composed of any number of States, and States can in turn be composed of any number of Tasks. This three-level structure provides a flexible vocabulary by which organizations store the history of the services that they provide to customers.

A Service may also be divided directly into tasks:



Services are defined by association to Service Types that you create as Business Attributes in the Configuration Layer. States may be used to represent components of customer service, such as:

- Customer identification
- Assigning a service agent (automated or live)
- Service identification
- Waiting for a service agent
- Offering another service while waiting for an agent
- Offering callback
- Waiting for customer input

Services, States and Tasks exist over some application-defined lifecycle. Upon completion, applications may specify a Disposition. For example, the offering of a new product or service might be recorded as a State of type Offer another service. The Disposition might be set to show whether the customer accepted or declined the offer. Information on past declined or accepted offers could then be used to calculate the likelihood that the customer might be interested in the offer at some point in the future.

### Important

This Service Model can be used by any component that can access UCS/CS's HTTP interface. It is not limited to use in Conversation Manager.

## Genesys Rules

Genesys Rules System provides the ability to develop, author, and evaluate business rules.

A **business rule** is a piece of logic defined by a business analyst. The rules in a rule package provide a set of functionality. The Genesys Rules Authoring Tool (GRAT) allows you to create, edit, and delete rules and rule packages.

**Rule packages** are bundles of rules. Rule packages are used to group, manage, and deploy rules. A rule package contains one or more rules plus the fact model that is needed to support the rules. The fact model is a description of the data. It contains field names and types which are grouped into tables/classes. Facts are input/output to rule execution and are instances of the tables/classes defined in the fact model. Rule packages also contain the rule definitions, business calendars, and also the templates that the rule package is dependent on. You deploy rule packages individually to the Rules Engine.

Rule packages also allow you to do the following:

- Partition rules and facts so that they are small, well-defined, and apply only to a particular application or use. This makes them easier to debug and understand.
- Isolate rule packages from one another when executing rules. This also improves performance because the Rules Engine has fewer candidates to examine during the evaluation.
- Update individual rule packages without affecting other deployed packages.
- Import and export an entire rule package containing the rule definitions, business calendars, and also the templates that the rule package is dependent on.

## Genesys Reporting

Run reports to determine customer trends.