

SimVentive

Rapid Development of Training Games and Simulations

Create Interactive Simulations Quickly and Easily

Traditionally, constructing simulations has been a software development task, requiring programmers to write C++ or Java code to define the behavior and appearance of the simulation. This limits the ability of instructional designers to create their own training games, and it increases the cost and time need to develop new simulations.

SimVentive™ is an innovative software suite that changes the simulation development process by enabling you to quickly and easily create simulations and games for training. Using the SimVentive Scenario Editor, you can rapidly construct a simulation scenario without programming. Scenarios can be played by the SimVentive Scenario Player, running as a Java application or as an applet to support both classroom instruction and distance learning.

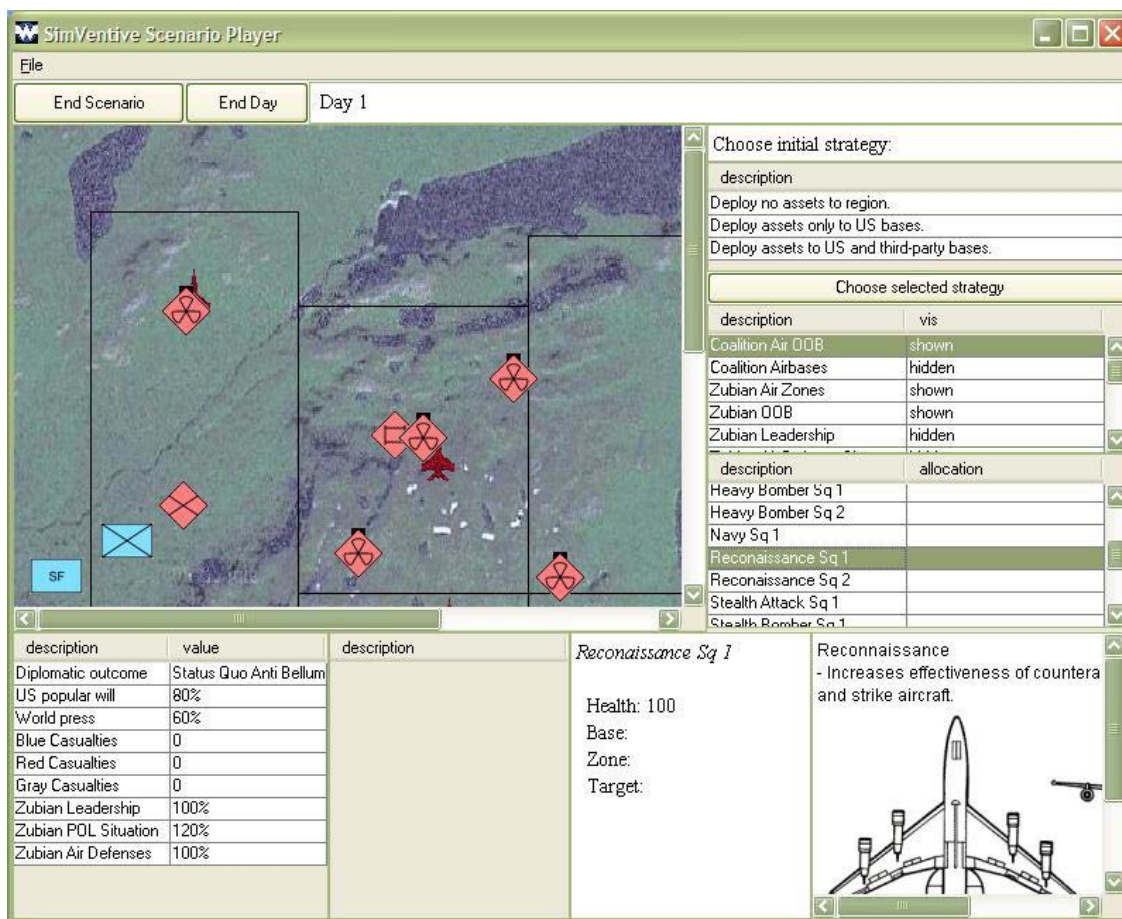


Figure 1 – The SimVentive Scenario Player presents the simulation's user interface using a variety of widgets.

Rapid User Interface Development

SimVentive's drag and drop UI Builder lets you create the user interface for your game by selecting, placing, and configuring user interface *widgets* such as images, web pages, maps, and tables. The UI Builder lets you set each widget's *properties* that specify its contents, appearance and interactive behavior. Each widget supports a number of *interactions* that describe how the widget communicates with the rest of the simulation when the user interacts with the widget. A button widget, for example, has one interaction, which is to be pushed. Using the UI Builder, you can configure this button widget's "push" interaction so that it sends an event to the rest of the simulation when the user clicks on it.

SimVentive provides a wide range of widgets allowing you to build sophisticated user interfaces for your games. The list of built-in widgets includes:

- **Image widget** - supports interactive "hotspots" that react to the user's mouse clicks and mouseovers. These interactive images enable the user to interact with pictures, screen captures, or schematics of people, places, software applications, equipment, or other simulated objects.
- **Map widget** - displays both bitmapped and tiled-based maps overlaid with user-defined drag and drop icons that represent simulated entities.
- **HTML widget** - displays HTML-formatted text, tables, and images with support for hyperlinks, input forms, and Cascading Style Sheets. The HTML can be specified at authoring time, loaded from web pages, or generated dynamically by simulation rules.
- **Dynamic table widget** - displays reports, statistics, and tables that reflect the current state of the simulation world.
- **Transcript widget** - displays a log of actions and dialog utterances by the simulation's characters and entities.
- **Standard user interface components** - such as push buttons, radio buttons, text entry boxes, and dialog boxes.

In addition, SimVentive enables you to extend the set of built-in widgets by integrating your own widgets using SimVentive's widget programming interface. With this API, you can easily create a wrapper for any existing Swing component to display it in your SimVentive scenario.

Development is underway on additional widgets to further extend SimVentive's capabilities. The DataMontage™ widget presents complex time-oriented data as arrays of timelines and graphs. The TaskGuide™ intelligent job-aid wizard provides step-by-step assistance guidance and user-system dialogs, and the interactive tutoring dialog widget supports automated instructional dialogs.

Rapid Definition of Simulation Objects

Behind the user interface for a SimVentive scenario lies the *simulation world*, which is made up of *entities* that represent all of the things being simulated, and their behaviors. Entities can be concrete – such as people, animals, cards, and telephones – or abstract –

such as organizations, facts, and decisions. The SimVentive Scenario Editor makes it easy to create and edit the different types of entities that you want to exist in your scenarios.

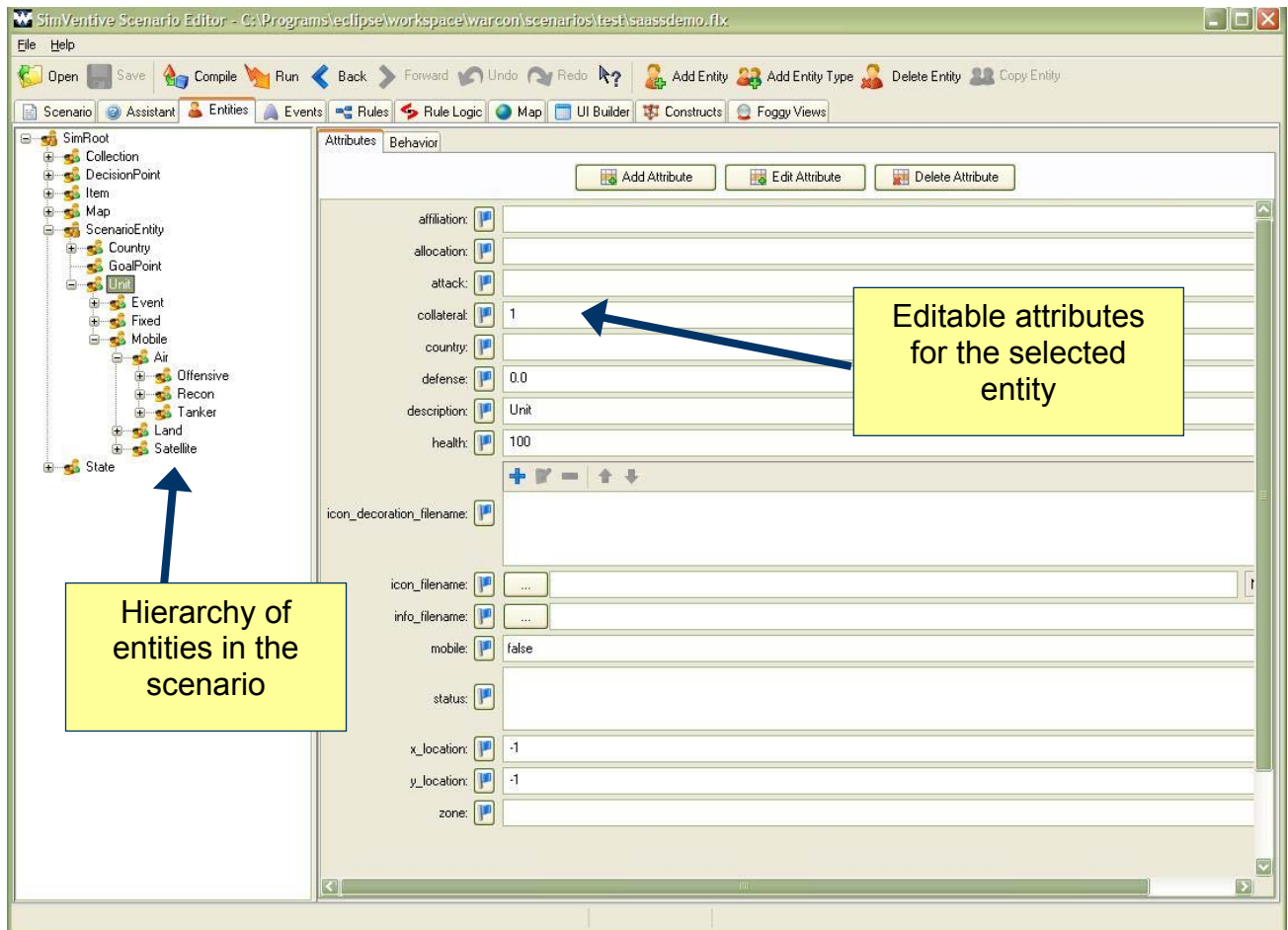


Figure 2 – The SimVentive Scenario Editor enables the author to create and edit simulation entities.

Each type of entity in a SimVentive scenario has a set of user-defined *attributes* that describe qualities of that entity such as its size, speed, friendliness, or cost. You can modify the value of an entity's attribute simply by typing the new value in the appropriate field.

The entities that you define in the Scenario Editor are tightly integrated with the scenario's user interface so that changes to an entity will be reflected in the UI. For example, if you add a map widget to your game's user interface, the map can automatically display the location of the entities in your scenario, moving their icons as the entities' locations change.

Rapid Development of Simulation Behavior

Rules define the logic behind the game by controlling how the simulation unfolds in response to player actions and by specifying how the entities in the scenario sense, reason, and act. Traditionally this would be done by writing C++ or Java code, but SimVentive instead provides a graphical rule editor (based on Stottler Henke's award-

winning SimBionic® intelligent agent development toolkit) that enables you to easily specify complex logic by drawing flowchart-like diagrams.

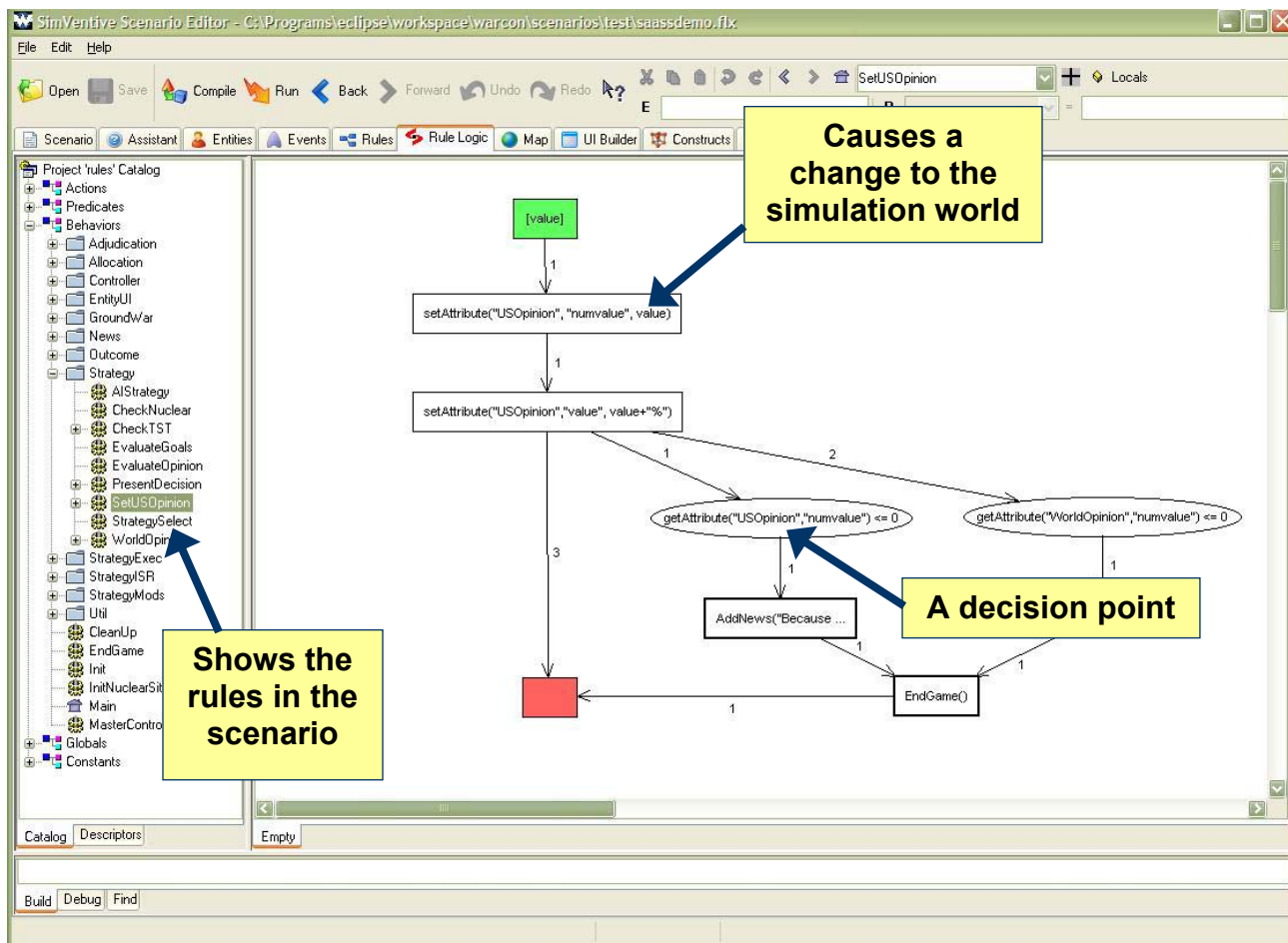


Figure 3 – The scenario author specifies simulation rules in the Scenario Editor’s Rule Logic pane.

Each SimVentive rule is made up of a sequence of actions and decisions, depicted above as rectangles and ovals. Actions change the state of the simulated world or the user interface – for example, by posting some information to an HTML widget to simulate an incoming news brief. Decisions are choices made based on the current state of the world or on interactions with the player. SimVentive rules give you access to the full power of SimBionic, enabling you to define simulation behavior that is as simple or as complex as you desire. Rules can be nested, so you can create modular, re-usable chunks of simulation logic. They can also be polymorphic, so the same rule will execute different actions and decisions depending on the attribute values of the entity that is running it.

In addition, SimVentive’s extensible architecture enables you to integrate arbitrary Java classes and methods into your rules to support sophisticated custom functionality. For example, you could create a SimVentive scenario that accesses your own proprietary business process model in order to increase the faithfulness and accuracy of the simulation.

Authoring Assistant Helps with Scenario Creation

The SimVentive Scenario Editor has an intuitive user interface designed to make scenario creation as simple as possible. Even expert authors get stuck sometimes, so the Scenario Editor also includes a wide range of features to help you through the process of creating scenarios. First and foremost is the Assistant, which analyzes your scenario-in-progress and points out any errors or possible mistakes that you may have made. It can also provide guidance on how to resolve these errors. In addition, the Assistant also provides interactive *wizards* that guide you through specific scenario-authoring tasks, such as creating a multi-player scenario.

The Scenario Editor also offers three different modes that cater to different types of users. The most basic mode, Edit, helps novice users quickly browse an existing scenario and make minor changes. It features a Parameters Editor that permits you to “tune” a scenario by changing key parameter values specified by the scenario author. The second mode, Build, is intended for users who wish to make significant changes to an existing scenario. It hides the most complex features of SimVentive but provides more powerful capabilities than Edit mode. Finally, Forge mode provides access to all of SimVentive’s capabilities. In this mode, users can create entirely new scenarios from scratch and change every aspect of a scenario.

SimVentive also provides the Scenario Debugger, a special version of the Scenario Player that displays helpful diagnostic information such as the state of the simulation world. Using the Debugger, you can quickly track down problems with the entities or rules in your scenario.

Powerful Event-Based Architecture

Behind the scenes, SimVentive is built on a powerful event-handling architecture. Everything that takes place in a scenario – including player interactions – is represented by an event. Scenario authors configure how rules and widgets react to each type of event. This event-based model makes SimVentive scenario authoring highly flexible and makes it easy to reuse pieces of old scenarios.

Events can also be scheduled to occur at particular times in the scenario. This can be useful for simulating events that always occur in the scenario regardless of what the player does. For example, you might define a “Hurricane” event that always occurs on the fifth turn of the scenario and causes oil prices to increase.

Support for Wide Range of Games

SimVentive supports a variety of different types of games and simulations:

- Turn-based (e.g. strategy games) games
- Continuous real time games
 - Complete control over the flow of time
- Single-player games

- Multiplayer games across a network, with support for:
 - Multiple teams
 - Multiple roles per team, with a custom user interface for each role

SimVentive also has built-in support for:

- “Fog of war,” where the player receives distorted or incomplete information about the world, as defined by the scenario author
- Dependencies between entities. This can be used to capture causal relationships between entities, such as “An increase in the price of oil will proportionally increase the shipping costs for product X.”

Reuse of Existing Game Content

Once you’ve created a SimVentive scenario, you can reuse parts of it in future scenarios to speed up the authoring process. Using the SimVentive Scenario Editor tools, you can build and manage a library of scenario modules and mix and match them to create new scenarios. Reuse rules, entities, or even entire pieces of the user interface from a past game, and spend your time working on new content rather than reinventing the wheel.

System Requirements

The SimVentive suite is 100% Java and is compatible with Windows 2000/XP, Linux, and MacOS 10 operating systems.

For more information about SimVentive, please contact Ryan Houlette at 617-616-1293 or houlette@stottlerhenke.com.