Stratmas Evolver User Guide

Revision : 1.5

Daniel Ahlin

January 31, 2007

Parallelldatorcentrum

1 Introduction and Terminology

The Stratmas evolver component aims to provide a way to automatically tune the various parameters and parts of a Stratmas scenario. Tuning implies a preferred state, that is the concept that some outcomes of a scenario is better than others. For instance, a plan consuming less resources is better than one that requires more (everything else equal). In order to decide whether the desired state is reached we need to be able to measure an observable performance criterion. In Stratmas this criterion is called the **target**.

Ideally we would like to be able to tune every available parameter in a given scenario to reach the optimum target. However, in practice this is not feasible due to the very rapid growth of complexity each additional tunable parameter causes (an effect known as the curse of dimensionality). Hence it is necessary to make a selection of exactly which parts of a scenario we would like to optimize, these will be called **parameters**.

The evolver works by making succeeding evaluations of the target for different parameter-settings. Each step in this chain is called an iteration and consequently there exist a count of iterations which will be called **evolver iterations**. In Stratmas the evaluation consists of simulating the scenario with the given parameter settings. Since the simulation is a stepwise procedure we also have a simulation iterations count, this will be called **simulation iterations**.

Several evolver iterations will naturally give raise to several target evaluations. The evolver will need a strategy to decide which of these it should consider to be the best, this will be called **search strategy**.

2 Requirements

The evolver needs additional resources to run, most of which should already be available in a working Stratmas environment. However, in the interest of completeness they are described here:

Scenario The purpose of the evolver is to develop and tune an already existing scenario. Consequently it needs a working scenario to use as a starting point.

Stratmas Server As described above, the evolver simulates a scenario in order to evaluate the target with particular parameter-settings. Hence the evolver needs access to at least one unoccupied Stratmas Server instance. (Please consult STRATMAS SERVER USER MANUAL on how to setup a Stratmas Server.)

Stratmas Dispatcher The evolver uses the Stratmas Dispatcher service to find available Stratmas Servers. Consequently it needs access to exactly one Stratmas Dispatcher instance. (Please consult STRATMASDISPATCHER - NOTES ON USAGE on how to setup a Stratmas Dispatcher and how to make the Stratmas Client and Server use it.)

3 Usage

This section is intended to give step-by-step instructions on how to evolve an existing scenario.

3.1 Accessing the Evolver

The evolver is accessed by choosing `Evolve Scenario` in the `Tools`-menu. (Please note that this item will be disabled if no scenario is opened or if there is no available Stratmas Dispatcher, see Section 2 above). This will cause a window resembling Figure 1 to be displayed (cosmetic details may vary between different platforms). This is the configuration window for the evolver, the next section will describe how it is used. Please note that the evolver works on a **copy** of the scenario. This means that no changes made to the scenario outside the evolver window will affect the scenario being evolved.

3.2 Configuration

As delineated in the introduction the evolver has several settings which decides the mode in which it will be operating. The aim of this section is to describe how these are specified.

3.2.1 Specifying the Target

The target is specified by dragging it from the tree to the left and dropping it to the right, in the blank area under "Target", denoted 2 in Figure 1. Currently supported target types are most number-valued parameters, and process variables. An error message will be delivered if an unsupported target type is dropped. To replace an already chosen target, just drop a new one in the target window.

Using a Process Variable as Target A process variable is quite meaningless without specifying the area it is supposed to be measured in. Reflecting this, the way of specifying a process variable as target is to drop a shape (such as, for instance, the map or its parts or the location of a specific unit) on the target field. Currently it is only possibly to use shapes already specified in the scenario.

Please note that in order to find out what process variables that are available the evolver will contact a Stratmas Server provided by the Stratmas Dispatcher. Consequently it is often necessary to have at least one server available even during configuration of the evolver.

3.2.2 Specifying the Parameter(s)

The parameters are specified in much the same way as the target, the main exception being that several parameters are allowed (whereas only one target is). The parameters are dragged from the tree to the left and dropped in the blank area under "Parameter", denoted 1 in Figure 1. Currently supported parameter types are most number-valued parameters. Again, an error message will be delivered if an unsupported parameter type is dropped.

3.2.3 Configure the Search Process

Search Strategy The currently available search strategies are "Minimize" and "Maximize", searching for lowest and highest target values respectively. The default is "Minimize" and can be changed to "Maximize" by clicking in the field denoted 3 in Figure 1.

Evolver iterations As described above, the evolver takes a series of steps, hopefully progressing to some optimum. The default number of steps to take is 10 and can be changed in the field denoted 4 in Figure 1.

3.2.4 Configure the Evaluation Process

As stated above the evaluation process of the evolver in Stratmas consists of simulating a scenario. The only configuration needed to accomplish this is to specify how many time-steps the simulation is supposed to take before the target is recorded. The default number is 10 and can be changed in the field denoted 5 in Figure 1.

3.3 Running and Inspecting

When the target and parameter(s) are specified the evolver is ready to run. This will be reflected by the run-button, 6 in Figure 1, changing from disabled to enabled. Once the evolver is started it is not possible to make any configuration changes. Make sure all desired configuration options are set and press the run-button (a confirmation dialog will be displayed).

When pressed, the run-button will change appearance to resemble a pause-button; if pressed again the evolver will pause when the current evaluation is finished (consecutive presses will toggle between the running and paused state). Please note that since the pauses are made *after* an evaluation is finished, there may be significant lag between the press to the pause button and the evolver actually pausing. When the evolver is finished the run-button will be disabled.

If the evolver is to be forcibly aborted, use the abort-button, 7 in Figure 1. Please note that the evolver can not be resumed after an abort.

When the evolver has started the evolver window will change appearance and contain two tabs, "Plot" and "Evaluations", described below.

3.3.1 Status Display

The status of the evolver, that is what it is currently doing, is displayed in the status bar, 8 in Figure 1.

3.3.2 Tabular presentation - The Evaluations Tab

The values of the parameters and target for every finished and running evolver iteration is available as a table in the Tab entitled "Evaluations". This tab is largely non-interactive but provides a rather complete view of the current knowledge of the evolver (meaning the currently available target-evaluations of sampled parameters).

3.3.3 Graphical presentation - The Plot Tab

Currently the evolver contains an experimental plotting facility aiming to (but currently not succeeding in) giving a feeling for the error landscape in which the evolver searches.

Navigation in this plot is made in so called first-person mode, using the scroll-wheel of the mouse to move into and out of the plot. Pressing the left button and moving the mouse rotates the viewer. Pressing the middle button and moving the mouse moves the viewer in the current plane of orientation.

To the right of the plot are four drop-down menus in which it is possible to select what values to use for respectively x-, y-, z-axis and color scheme. Below these are several experimental visualization options, allowing the user to turn on and of various features.

Please note that the details of this plot is likely to change in the future.

3.4 Evaluating the Result

When using results derived from machine learning it is always important to check if the given results are reasonable. Since the Stratmas evolver currently has very little subject matter knowledge, this is especially important when using results from this implementation. It is strongly recommended to investigate the results from the different iterations using the tabular view to get a feeling for how the evolver found the end result.

It should also be noted that since the Stratmas evolver currently uses an iteration-count stop criterion (meaning that it stop after a certain number of iterations, no matter what), the result of the last iteration is by no means guaranteed to be the best.

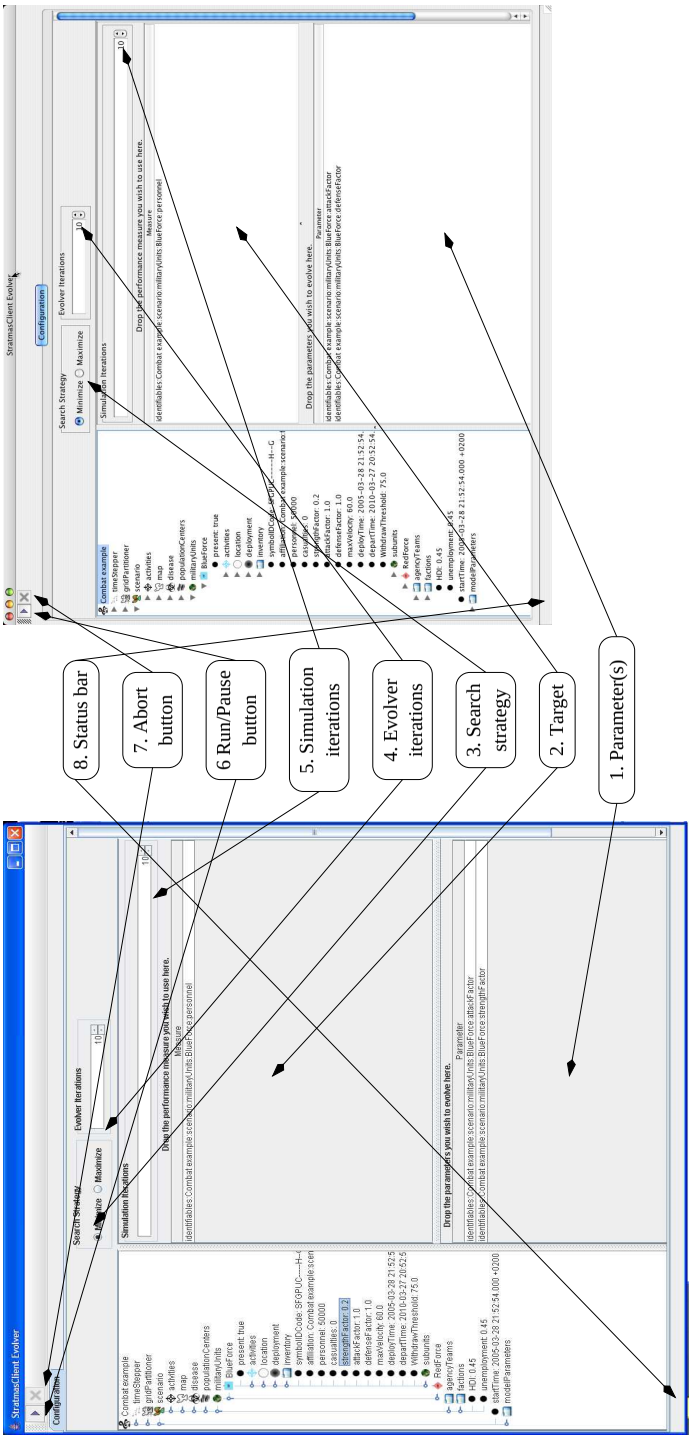

Figure 1. Configuration Window A view of the evolver configuration window on Windows XP and Mac OS 10.4. Please refer to section Section 3.2 for a description of the various settings.