



A1.D4.1 – Preliminary version of the static analysis tool

A. Bonetti, A. Saidane

Document Number	A1.D4.1
Document Title	Preliminary version of the static analysis tool
Version	1.3
Status	Final
Work Package	WP 1.4
Deliverable Type	Prototype
Contractual Date of Delivery	31 December 2007
Actual Date of Delivery	09 February 2007
Responsible Unit	UTN
Contributors	UTN
Keyword List	
Dissemination level	PU

Change History

Version	Date	Status	Author (Unit)	Description
1.0	5/2/2007	Working	A. beonetti (UTN)	Initial input
1.1	5/2/2007	Working	A. Saidane (UTN)	Minor corrections
1.2	9/2/2007	Final	A. Saidane (UTN)	Minor corrections
1.3	9/2/2007	Final	A. Saidane (UTN)	Minor corrections

Executive Summary

The document contains the user's guide for the Preliminary version of the static analysis tool.

Table of Contents

1. Introduction.....	5
1.1. Introduction	5
1.2. Required System configuration	5
1.3. Installation.....	5
1.2.1 Compile the sources.....	5
2. Drawing a model.....	6
2.1. Model’s elements.....	6
2.2 Actors	7
2.3 Services	8
2.4 Ternary relations.....	8
3. Formal Analysis.....	9
3.1. Analysis.....	9
3.2 Goal Reasoning Analysis	9
3.3 Manage Scenarios.....	10
3.4 Datalog Analysis.....	10
3.5 Specifying Datalog options	11
4. Main options.....	12
4.1. Export model	12
4.2 Manage model’s colour.....	12

1. Introduction

1.1. Introduction

Serenity Tool is a CASE tool developed for modelling and verifying functional and security requirements. Main goals of the tool are:

- Graphical environment: a visual framework to draw functional and security requirements;
- Formalization: support to translate visual models into formal specifications;
- Analysis capability: a front-end to external tools for formal analysis.

Serenity Tool is mainly composed of two parts: Serenity Tool kernel and external solvers. Serenity Tool kernel has an architecture comprised of three major parts, each of which is comprised of modules.

1.2. Required System configuration

Serenity tool is a Swing-based Java application that supports only Win platform. It requires a Java virtual machine - v.1.4 or above - properly installed to run. Also the external solvers are included in the Serenity tool package.

1.3. Installation

Serenity Tool 1.0 comes in a zipped package. It does not need special installation: just unzip the package into a directory and double-click the .jar file. Otherwise the tool can be run from the shell using the following command:

```
java -jar SERENITY_Tool_1_0.jar
```

1.2.1 *Compile the sources*

Serenity tool can also be build from the sources, to do this just follow these steps:

1. Unzip the Serenity tool package
2. `cd [serenity tool dir]\src`
3. `compile.bat`
4. `cd [serenity tool dir]\classes`
5. `java grtool.Application`

2. Drawing a model

2.1. Model's elements

Models are drawn by selecting entities from the tools bar and placing them on the working area. Supported entities are:

- Actors
 - Actors;
 - Agents;
 - Roles;
- Relations
 - Decomposition
 - Contribution
 - Association: is-part-of, is-a, play, supervise
 - Delegation: delegation of execution, delegation permission.
 - Trust: trust execution, trust permission, trust monitor.
 - Monitoring: monitoring permission, monitoring execution.
 - Ownership.
 - Request.
 - Provide.
- Services
 - Goals;
 - SoftGoals;
 - Tasks;
 - Resources;
 - Events;

Some usage tips:

- Hold down Ctrl key for multiple insertions.
- Insert a service placing it over another Actor to automatically set the service's belonging to the actor.
- Insert a service placing it over another service to automatically create a composition relation between them

2.2 Actors

When you insert an actor in the diagram, it's initially collapsed: its rational is hidden and it appears as a simple circle (or a different simple symbol, depending if it is an actor, agent or role). To expand or collapse it again, select the actor, and check/uncheck the "collapsed" button in its property page. It is also possible to collapse it by double-clicking on the actor's shape. Placing new services on an actor's rational means that you assign them to it.



Figure 1 – Actor's type

If the actor is collapsed again, the visibility of its nodes and relations is computed according to some rules:

- *non-dependum* services are hidden;
- inner relations are hidden;
- incoming/outgoing relations are automatically updated to maintain model's consistency.

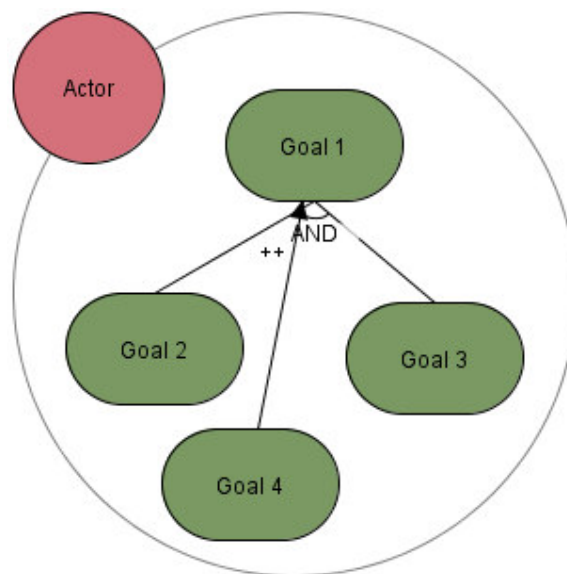


Figure 2 – Actor's rational

2.3 Services

Inserting a service over another service also establishes a decomposition relation between the new and the existing service. Then, drag the new service to place it at the right position.

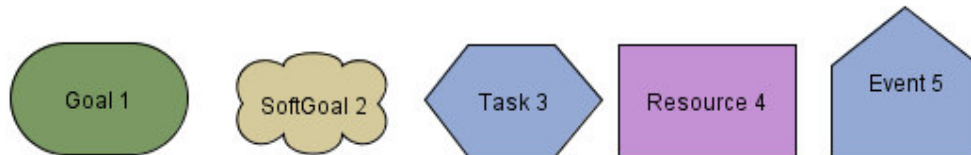


Figure 3 – Service's type

2.4 Ternary relations

Delegation, Trust, Monitor: all of these are ternary relations; they involve a *depender* actor, a *dependum* service and a *dependee* actor. To draw such relation, you can use an existing service or just connect two actors. There are two major cases:

- the *dependum* belongs to an actor's rational;
- the *dependum* doesn't belong to any actor.

In the first case, the relation is represented by a single arc. In the second case, there are two relation arcs: one from the depender to the dependum, and another from the dependum to the dependee.

3. Formal Analysis

3.1. Analysis

Serenity Tool is able to manage two different kind of formal analysis: the Goal-Risk Reasoning and the Datalog Analysis. In both situations Serenity Tool used two external solver, in the first case the “GoalSolve” solver, and in the second case the DLV solver.

3.2 Goal Reasoning Analysis

Serenity Tool allows to perform a Bottom Up Goal-Risk Analysis using the external solver “GoalSolve”. The tool shows the results of the analysis in two way:

- Labelling every service that is related to an actor with two character, one for the Satisfied value and the other for the Denied value. These values can be:
 - “T”: for the total satisfied/denied value.
 - “P”: for the partial satisfaction/denied value.
 - null : for the null satisfaction/denied value.
- Colouring every services related to an actor with a different color that represent the satisfied/denied value or the presence of weak/strong conflicts:
 - Green: it means that the service is fully satisfied.
 - Yellow: the service is partial satisfied.
 - Orange: the service is partial denied.
 - Red: the service is fully denied.
 - Light grey: weak conflict, the service is partial satisfied and partial denied at the same time.
 - Dark grey: strong conflict, the service is totally satisfied and totally denied at the same time.
 - White: no value related to the service.

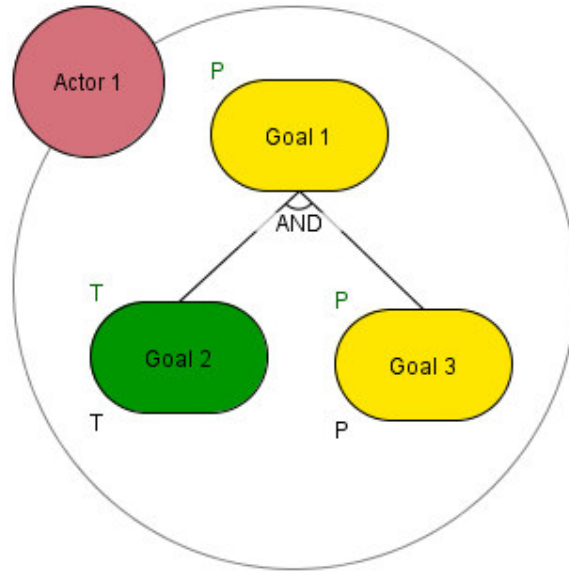


Figure 4 – GR Analysis results

3.3 Manage Scenarios

Every model can have different “Scenario”, this means that different values of satisfied and denied can be related to the same service. This can be helpful to understand better the relations between services. The management of the different scenarios can be made using the second tab called “GR-Options”.

3.4 Datalog Analysis

Serenity Tool use DLV solver to perform formal analysis on Model. The tool package include the binaries of DLV solver and everything is ready to use. To perform the formal analysis just click on the related button, to clear the results just pressed the other button.

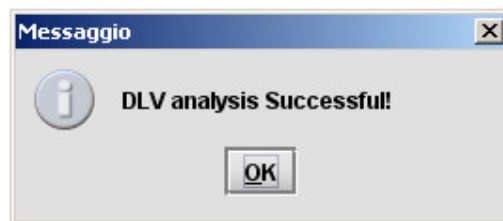


Figure 5 – A successful DLV analysis

In case of successful analysis a Panel appears indicating that the model present no problems. If the solver encounter problems in the analysis of the model the tool will show the problems in the model using red arrows. Every arrow is labelled with the problem that it shows and with the number of the solutions to which is related.

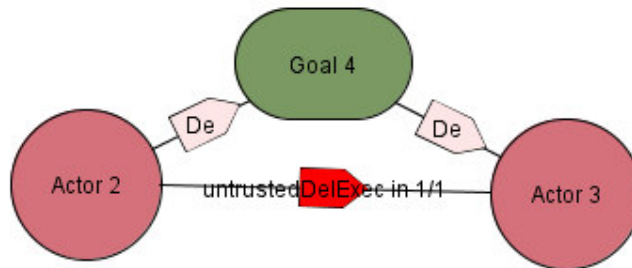


Figure 6 – Some model's errors from DLV analysis

3.5 Specifying Datalog options

Choose properties. Here you can select entire families of security properties to verify.

Add files with further specifications. Here you can add your own files to be passed to the solver. Files are passed to the solver "as they are", without any verification.

Use conditions? In the graph editing panel, in the "datalog" tab of each relation you can specify a condition for the relation. Here, you tell to the analysis system to include such conditions (when present) in datalog encoding, or to exclude them.

4. Main options

4.1. Export model

Serenity tool allows to export the model in two different file format:

- .eps, Encapsulated PostScript.
- .png, Portable Network Graphics.

These operation can be managed using the related button in the tool bar.

4.2 Manage model's colour

Different Colour Schema are available for drawing models:

- Default, that use light colours.
- Taom4E, that use darker colours.
- Black\&White.

The colour schema can be changed using the Colour button in the tool bar.