SysADL Studio

User Manual

Eduardo Silva

November 6, 2019

Version Control

Version	Date	Author	Editor	Description
0.1	Sept 2019	Eduardo Silva	Eduardo Silva	Initial Version

Contents

Chapter 1

Overview

This manual describes SysADL Studio and its features, focusing on how the modeling is performed with this tool support. Further information about SysADL can be found in `http://sysadl.org`, the official page of the language.

1.1 About SysADL

SysADL is an Architecture Description Language designed based on the OMG SysML [1]. The language encompasses a set of complimentary views that supports architectural modeling from the conceptual, high-level perspective to the lower, close to implementation, level. The language is under constant validation, receiving feedback provided by the industry and students of Brazilian and French universities.

The language plays a key role in the book [], in which it is used to illustrate software architecture concepts.

1.2 Overall Information

SysADL Studio is a toolset that encompasses four plugins, in which the core plugin is the only mandatory to supports SysADL modeling. Each additional plugin provides a set of features that extends the core functionalities providing a complete environment that supports modeling, both textual and graphic, verification and execution.

The toolset was build using a set of modern, consolidated frameworks. Specifically, we combine EMF, Sirius and Xtext to provide a complete modeling environment. However, the tool is not restrained to those frameworks, additional plugins enhance the user experience and provide useful tools for architectural modeling.

[1] `http://sysml.org`

Plugin	Version	Description
SysADL Core	2.7.0	Core Plugin
SysADL Viewpoints	2.7.2	Enables Graphical Modeling of SysADL
SysADL Verification	1.0.3	Supports Model Verification over Styles and provide tools for calculating architectural metrics
SysADL Execution	0.5.qualifier	Enables Model Execution (not available for download)

Table 1.1: SysADL Plugin Ecosystem

1.3 Plugins and Features

Table 1.1 summarizes the plugins, presenting their current versions and the feature it adds to the environment. Please note that the presented versions correspond to the date of the publication of this manual, they might vary with no impact in the features hereby described.

1.3.1 SysADL Core

SysADL Core is the plugin responsible for allowing creation and manipulation of SysADL Models, it is the main plugin in the infrastructure of SysADL Studio and its installation is always mandatory. The plugin provides a textual editor for SysADL, and also an extensible infrastructure over which SysADL Studio's environment is built.

1.3.2 SysADL Viewpoints

SysADL Viewpoints is the plugin responsible for providing an additional, graphical, mean of creating, manipulating and visualizing SysADL Models. It enhances the environment by adding a set of viewpoints that encompasses the SysADL diagrams.

1.3.3 SysADL Verification

SysADL Verification is an extension plugin for SysADL Viewpoints, it enhances the diagrams to support verification of constraints described in OCL. These constraints can be defined in a *Style* and are verified when applied to an *Architecture*.

Further, the verification plugin also provides the user a set of tools to calculate architecture metrics, such as cohesion and coupling.

1.3.4 SysADL Execution

SysADL Execution extends the environment by adding a Debug Perspective that allows SysADL models to be executed and hence simulated. The plugin encompasses a powerful execution engine that understands SysADL syntax and implements its semantics, allowing the user to observe the execution step-by-step.

Chapter 2

Quickstart

This chapter provides the basics understanding about the more trivial processes in SysADL Studio: the Installation and creation of our first SysADL Model.

2.1 Installation

2.1.1 Prerequisites

We recommend to install SysADL Studio in the latest version of Eclipse. We also recommend that the installation is done on the Modeling distribution, since SysADL Studio has several dependencies that are met by this distribution and may be absent in other packages.

By the date of this manual, the most recent version is Eclipse 2019-09. Eclipse can be downloaded in `https://eclipse.org`, Modeling distribution can be found in `https://www.eclipse.org/downloads/packages/`.

2.1.2 Installing SysADL Studio

To install SysADL Studio you will need to fulfill the prerequisites described in Section 2.1.1.

The installation process is simple, find the menu Help ¿ Install New Software on your Eclipse, as presented by Fig. 2.1.

The SysADL Update Site is the repository in which our plugins are stored, the page is located at: `http://sysadl.org/update`. It is important to highlight that this page **is not** a website, hence it **can't be opened by a web browser** such as Microsoft Edge, Google Chrome or Mozilla Firefox.

Type the SysADLStudio update site in the *Work With* box, as presented in Fig. 2.2 and press enter, Eclipse will then load the update site.

Figure 2.1: Install New Software Menu

Figure 2.2: Inserting the update site

Figure 2.3: Selecting Plugins for installation

Select the desired plugins (potentially all of them), as illustrated by Fig. 2.3 and follow the instructions to conclude installation. During the installation, Eclipse will warn you about the unsafety of the update site, **this is an expected behavior and it is normal**, since SysADL Studio is not hosted in `http://eclipse.org`.

After the installation, it is necessary to restart your IDE, do it as soon as possible. SysADL Models and tools will be available next time your IDE starts.

2.2 First SysADL Model

To work with SysADL it is necessary to create a SysADL Model. These models can be created in any project, although we recommend the use of **Modeling Projects**.

First, create a project (preferred a Modeling Project) and name it as desired. Then, create a SysADL Model, as illustrated by Fig. 2.4. SysADL Textual Editor can be used to edit these models and it is further described in Chapter 5.

After creating a SysADL Model it is possible to edit graphically using the SysADL Editor. However, if willing to use the graphical editor, it is important to ensure your IDE has the *SysADL Diagrams* plugin installed.

If the required plugin is present, it is possible to associate graphical viewpoints to edit the SysADL Model. This requires Sirius Perspective, that can be selected using the Change Perspective menu, located in top-right of your IDE, as shown in Fig. 2.5. Open the menu and select Sirius, your IDE will reconfigure to the adequate environment.

Whilst in Sirius Perspective, a right click in a project that contains a SysADL Model allows to Select Viewpoints, as illustrated by Fig. 2.6. Select the desired SysADL Viewpoints and press *Ok*. It is important to notice that besides an initial processing, the number of selected viewpoints does not interfere in the performance of your IDE.

Viewpoints selected it is possible to navigate through the diagrams using the *Model Explorer*, using a tree structure in this navigation unit, as shown by Fig. 2.7. Any SysADL diagram can be accessed through this method, although some can be easier accessed by a in-diagram navigation feature. Further information about overall diagrams editors are introduced in Chapter 3.

Figure 2.4: New SysADL Model

Figure 2.5: Changing Perspective

Figure 2.6: Viewpoint Selection

Before start editing your SysADL Model, it is important to highlight that SysADL Models come with an default package, named *SysADL.types*. This package **should not be edited** and contains a set of basic types SysADL Studio understands.

Figure 2.7: Model Explorer

Chapter 3

Viewpoints and Associated Features

This Chapter describes the SysADL Viewpoints, that are provided by the plugin *SysADL Diagrams*.

3.1 SysADL Viewpoints

SysADL encompasses a set of 4 views, SysADL Studio provides a viewpoint for each of these views and an additional viewpoint to support verification of properties at design time. These viewpoints are described in this Section.

Each viewpoint introduces a set of diagrams that supports the visualization, creation and manipulation of the architectural model through the viewpoint's perspective.

Styles and Verification Viewpoints add features to existing diagrams, as Block Definition Diagrams (Structural and Behavioral).

View	**Viewpoint**	
Structural	Structural	Styles
Behavioral	Behavioral	
Execution	Execution	
Requirements	Requirements	
-	Allocation	
-	Verification	

Table 3.1: SysADL Views and Viewpoints

Viewpoint	**Diagrams**
Structural	Package Diagram Block Definition Diagram (Structural) Internal Block Diagram
Behavioral	Block Definition Diagram (Behavioral) Activity Diagram Parametric Diagram
Execution	Block Definition Diagram (Execution)
Requirements	Requirements Diagram
Allocation	Allocation Table
Styles	Block Definitino Diagram (Styles)
Verification	-

Table 3.2: SysADL Diagrams

3.2 Model Navigation

3.3 Diagram Tools

3.3.1 User Palette

3.3.2 Adding Notes

3.3.3 Exporting to Image

Chapter 4

SysADL Diagrams Usage

4.1 Structural Viewpoint

4.1.1 Package Diagram

4.1.2 Block Definition Diagram (BDD)

4.1.3 Internal Block Diagram (IBD)

4.2 Behavioral Viewpoint

4.2.1 Block Definition Diagram (BDD)

4.2.2 Activity Diagram

4.2.3 Parametric Diagram

4.3 Execution Viewpoint

4.3.1 Block Definition Diagram (BDD)

4.4 Requirements Viewpoint

4.4.1 Requirements Diagram

4.5 Allocation Viewpoint

4.5.1 Allocation Table

4.6 Styles Toolkit

4.6.1 Defining Architectural Styles

4.6.2 Using Architectural Styles

Chapter 5

Textual Editor

5.1 Synchronism with Diagrams

5.2 Features

5.2.1 Content Assistant

5.2.2 Formatter

Chapter 6

Executing SysADL Models

6.1 Prerequisites

6.2 Execution Environment

6.3 Interacting with the Model

Chapter 7

Verification Tools

7.1 Diagram Verification

7.2 Textual Verification

7.3 Model Verification

Chapter 8

Frequently Asked Questions