Grant Agreement: 287829

Comprehensive Modelling for Advanced Systems of Systems

Simulator/Animator Design Document

Technical Note Number: DXX

Version: 0.1

Date: Month Year

Public Document

http://www.compass-research.eu

11 ## Contributors:

12 Anders Kaels Malmos, AU

13 ## Editors:

14 Peter Gorm Larsen, AU

15 ## Reviewers:

Document History

Ver	Date	Author	Description
0.1	25-04-2013	Anders Kaels Malmos	Initial document version

Abstract

This document describes the overall design of the CML simulator/animator and provides an overview of the code structure targeting developers.

Contents

1 Preface

This document describes the overall strucure and design of the CML simulator, it is not a detailed description of each component. This kind of documentation is done in java doc and can be generated automatically from the code.

2 Overall Structure

This section describes the overall source code structure of the CML interpreter.

The CML interpreter is implemented in two separate components: a core component and an IDE component. The core component implements the operational semantics that are defined in D23.2 and is located in the java package named *eu.compassresearch.core.interpreter*. The ide component exposes the core component to the Eclipse framework as an integrated debugger. It is located in the *eu.compassresearch.ide.cml.interpreter_plugin* package.

2.1 The Core Structure

The following two packages defines the top level structure of the

eu.compassresearch.core.interpreter This package contains all the classes and interfaces that defines the core functionality of the interpreter.

eu.compassresearch.core.interpreter.api This package contains all the public classes and interfaces that defines the API of the interpreter.

The reason for this top level structure is to encapsulate all the classes and interfaces that makes up the core functionality of the interpreter and only expose the classes and interfaces that are needed to utilize it. This provides a clean separation between the implementation and the public interface.

The eu.compassresearch.core.interpreter package are split into several folders, each representing a different logical component. The following folders are present

cml

visitors

util

debug

...

2.2 The IDE Structure

3 Simulation/Animation

This section describes the static and dynamic structure of the component involved in simulating a CML model.

3.1 Static Structure

The top level interface of the interpreter is depicted in figure 1

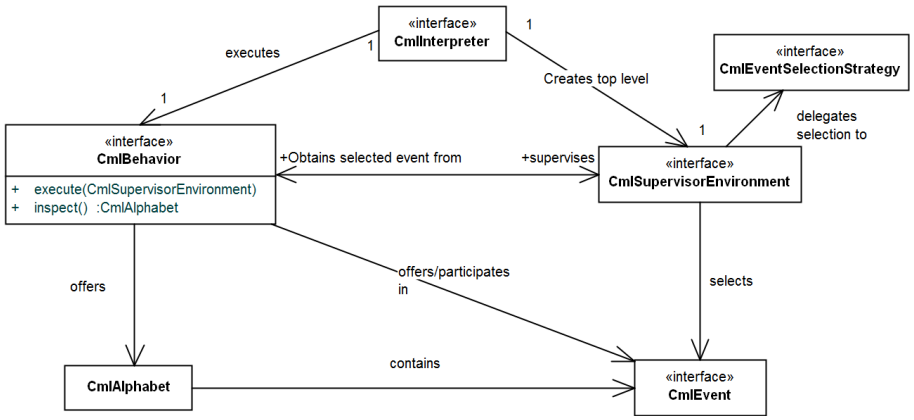

Figure 1: Diagram depicting the high level design of the interpreter core component

CmlInterpreter The main interface exposed by the interpreter component. This interface has the overall responsibility for interpreting. It exposes methods to execute, listen on interpreter events and get the current state of the interpreter. It is implemented by the **VanillaCmlInterpreter** class.

CmlBehaviour Interface that represents a behaviour specified by either a process or action. It exposes two methods *execute* which performs the behaviour and *inspect* which returns the immediate alphabet of the behaviour. A specific behaviour can for instance be the Skip action, which when executed successfully terminates the action.

CmlSupervisorEnvironment Interface with the responsibility of acting as the environment for processes and actions. This involves choosing the next event (if any) given the available events. It has a method for retrieving the occurring event.

CmlEvent Interface that represents any kind of event. This structure will be described in more detail in section 3.1.1

CmlAlphabet This class is a set of CmlEvents.

CmlEventSelectionStrategy This interface has the responsibility of choosing one event from a CmlAlphabet

3.1.1 Event Structure

3.2 Dynamic Structure