## **Artifact Availability**

Eclipse products for the Windows and Linux platforms into which OSATE and OSATE-DIM have been installed are provided in a Zenodo repository (DOI: 10.5281/zenodo.6971721). In the provided archive file, the *eclipse* directory contains the Eclipse product and the *workspace* directory contains our test project as an artifact supporting the claims in the paper.

The steps to set-up the OSATE-DIM testing environment are as follows:

- 1. Unzip the archive.
- 2. Initialize the Eclipse IDE by clicking on the 'osate' executable file within the osate directory of the archive.
- 3. You will be prompted by the OSATE2 Launcher, to select a workspace: Please select the *workspace* directory which is provided within the archive.

| OSATE2 Launcher  |  |          |  |  |  |  |  |  |  |  |
|--|--|----------|--|--|--|--|--|--|--|--|
| Select a directory as workspace<br>OSATE2 uses the workspace directory to store its preferences and development artifacts. |  |          |  |  |  |  |  |  |  |  |
| <u>W</u> orkspace:   | osate2-2.11.0-vfinal-linux.gtk.x86_64/workspace      | ▼ Browse |  |  |  |  |  |  |  |  |
| Use this a   | s the default and do not ask again<br><b>kspaces</b> |          |  |  |  |  |  |  |  |  |
|  |  | Cancel   |  |  |  |  |  |  |  |  |

4. In the Eclipse IDE, within the AADL Perspective, you will import the test project by clicking 'Import Projects...' in the AADL Navigator or by clicking menu 'File>>Import...'.

In the Import wizard, select 'General >> Projects from Folder or Archive'. Click 'Next'.



5. In the 'Import Projects from File System or Archive' dialog, specify the Import source as a Directory. The directory is contained within the workspace directory. It will look like this:

| Import Projects from File System or Archive 🛛 🛛 😣  |   |                           |                |                    |                  |  |  |  |  |  |
|--|---|---------------------------|----------------|--------------------|------------------|--|--|--|--|--|
| Import Projects from File System or Archive           This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE. |   |                           |                |                    |                  |  |  |  |  |  |
| Import source:   | 0-vfinal-linux.gtk.x86_64/workspace/        | fr.mem4csd.osatedim.tests |                | Di <u>r</u> ectory | <u>A</u> rchive  |  |  |  |  |  |
| type filter text   |   |                           |                | Sel                | ect All          |  |  |  |  |  |
| Folder   |   | rt as                     | Deselect All   |                    |                  |  |  |  |  |  |
| ✓ fr.mem4cs  | ✓ fr.mem4csd.osatedim.tests Eclipse project |                           |                |                    | dy open projects |  |  |  |  |  |
| Search for ne<br>Search for ne<br>Detect and co<br>Working sets  | sted projects<br>nfigure project natures    |                           |                |                    |                  |  |  |  |  |  |
| Add projec   | to working sets                             |                           |                |                    | Ne <u>w</u>      |  |  |  |  |  |
| Working sets:  |   |                           |                | •                  | S <u>e</u> lect  |  |  |  |  |  |
|  |   |                           | <u>Show ot</u> | her specialize     | d import wizards |  |  |  |  |  |
| ?  |   | < Back Next               | >              | Cancel             | Finish           |  |  |  |  |  |

- 6. Click 'Finish'.
- 7. You will now see the 'fr.mem4csd.osatedim.tests' project within the AADL Navigator.

## **Reproducing Results**

1. Open the Plug-in Development Perspective by going to 'Window >> Perspective >> Open Perspective'. Select 'Plug-in Development' in the 'Open Perspective' dialog.

| Open Perspective 🛛 🛛 😵  |  |  |  |  |  |  |  |  |
|---|--|--|--|--|--|--|--|--|
| & AADL (default)  |  |  |  |  |  |  |  |  |
| 🎋 Debug   |  |  |  |  |  |  |  |  |
| 🔚 Git   |  |  |  |  |  |  |  |  |
| 🖏 Java  |  |  |  |  |  |  |  |  |
| 🔊 Java Browsing   |  |  |  |  |  |  |  |  |
| 🖫 Java Type Hierarchy   |  |  |  |  |  |  |  |  |
| e <sup>™</sup> Modeling                                       |  |  |  |  |  |  |  |  |
| () Planning   |  |  |  |  |  |  |  |  |
| 🐠 Plug-in Development   |  |  |  |  |  |  |  |  |
| Resource  |  |  |  |  |  |  |  |  |
| Scripting   |  |  |  |  |  |  |  |  |
| 🖆 Team Synchronizing  |  |  |  |  |  |  |  |  |
| V Transformation Development                                  |  |  |  |  |  |  |  |  |
| Use F2 to display the description for a selected perspective. |  |  |  |  |  |  |  |  |
| Cancel Open   |  |  |  |  |  |  |  |  |

- 2. Unfold the 'fr.mem4csd.osatedim.tests' project and observe its structure:
  - ▶ ➡ JRE System Library [JavaSE-1.8] ▶ 
    <u>►</u> Plug-in Dependencies SIC # fr.mem4csd.osatedim.tests 🔻 🗁 cases experiment 🕨 🗁 mc-dag 🕨 🗁 ramses-ref-linux AADL Runtime.aadl 🖹 RAMSES.aadl 🕨 🗁 diagrams META-INF run-configs TestDeltaInplace.testRAMSES.launch TestState.testMCDAGState.launch TestState.testRAMSESState.launch a build.properties 🖻 representations.aird
    - a. The src/ package contains the source code responsible for interfacing the examples with the OSATE-DIM plug-in. It contains the classes and methods which perform the changes/refinements on the instance models. Users can observe each particular change and its program within this package, to also see how OSATE-DIM can be interfaced with other AADL-based tools.

- b. The cases/ directory contains the .aadl (Declarative) and .aaxl2 (Instance) models that are used in the case-studies. The models are contained in specific subdirectories according to the case-study (MC-DAG, RAMSES, Experiment) and scenario (state-based, delta-inplace, delta-outplace)
- c. To run the tests, go to the 'Run >> Run Configurations' menu : In the 'Run Configurations' dialog box, you will see the JUnit tests associated with OSATE-DIM

|  | Run Configura  | tions                               | 0 🔇    |  |  |  |  |  |  |  |  |  |
|--|--|-------------------------------------|--------|--|--|--|--|--|--|--|--|--|
| Create, manage, and run configurations Create a configuration that will launch a JUnit test. |  |                                     |        |  |  |  |  |  |  |  |  |  |
| 🖺 🖻 🕼 🗶 🖻 🏹 🕶 Name: TestState.testMCDAGState   |  |                                     |        |  |  |  |  |  |  |  |  |  |
| type filter text   | 🗉 Test 🕬= Arguments 🚘  | ironment »1                         |        |  |  |  |  |  |  |  |  |  |
| EASE Script  | Run a single test  |                                     |        |  |  |  |  |  |  |  |  |  |
| Eclipse Application  | Project:   | fr.mem4csd.osatedim.tests           | Browse |  |  |  |  |  |  |  |  |  |
| Java Application   | Test class:  | fr.mem4csd.osatedim.tests.TestState | Search |  |  |  |  |  |  |  |  |  |
| ✓ JuJUnit<br>JuTestDeltaInplace.testRAMSES   | Test method:   | testMCDAGState                      | Search |  |  |  |  |  |  |  |  |  |
| J TestState.testMCDAGState<br>J u TestState.testRAMSESState                                  | Run all tests in the selected project, package or source folder: |                                     |        |  |  |  |  |  |  |  |  |  |
| JU JUnit Plug-in Test  | fr.mem4csd.osated  | Search                              |        |  |  |  |  |  |  |  |  |  |
| OSGi Framework   | Include and exclude tags:  |                                     |        |  |  |  |  |  |  |  |  |  |
|  | Test runner:   | JUnit 4 🔹                           |        |  |  |  |  |  |  |  |  |  |
|  | Keep JUnit running after a test run when debugging               |                                     |        |  |  |  |  |  |  |  |  |  |
| Filter matched 11 of 11 items  |  | Show Command Line Revert            | Apply  |  |  |  |  |  |  |  |  |  |
| ?  |  | Close                               | Run    |  |  |  |  |  |  |  |  |  |

d. Select one of them and Click 'Run'

## **Observing the Results**

1. After running a particular test, observe the newly created and modified files through the Project Explorer view (in the 'Plug-in Development' perspective) or the 'AADL Navigator' view (in the AADL perspective).

| <ul> <li>build.properties</li> <li>representations.aird</li> </ul>  | <ul> <li>TestState.testMCDAGState.launch</li> <li>TestState.testRAMSESState.launch</li> </ul>   | TestDeltaInplace.testRAMSES.launch   | Se META-INF     Se run-configs  | diagrams  | AADL_Runtime.aadl  | <ul> <li>Construction</li> <li>Construction</li> <li>Construction</li> </ul>   | 🛱 declarative_null.aadl   | Solution declarative_main_impl_Instance.aaxl2 | ✓ ➢ instances ☆ declarative main impl Instance test.aaxl2   | ► 🦢 deta-outplace<br>► 🍅 state   | <ul> <li>Belta-inplace</li> </ul>  | • Sexperiment                                    | <ul> <li># fr.mem4csd.osatedim.tests</li> <li>Cases</li> </ul> | ▼ BSC       | Plug-in Dependencies    | JRE System Library [JavaSE-1.8]  | ▼ I fr.mem4csd.osatedim.tests         | 🎦 Project Explorer 🗙 💲 Plug-ins 📄 💲 🍸 🖇 😑 🗎 |
|---|---|--|---|---|--|--|---|---|---|--|--|--|--|-------------|-------------------------|--|---------------------------------------|---|
| RAMSES properties::Execution Slots => ([Computation Unit => reference (cpu.corel); Start Time => 900ms;<br>End_Time => 1100ms;]) in modes (LO), ([Computation Unit => reference (cpu.corel);<br>Start_Time => 1100ms; End_Time => 1000ms;]) in modes (HI) applies to proc.tg_phasel.Stab;<br>end main.impl; | End Time => 300ms;)) in modes (HI), ([computation_unit => reference (cpu.core1); Start_Time => 0ms;<br>End Time => 300ms;)) in modes (HI), ([computation_unit => reference (cpu.core1); Start_Time => 0ms;<br>End_Time => 300ms;)) in modes (LO) applies to proc.tg_phase1.Avoid; | nwrsts properites::txecutum sous -> (tomputentum nutter) = reterence (cputoreit); start Time -> 1sooms;<br>End Time => 160ms;]) in modes (LD) applies to procitg phasel.Com;<br>BMMSES properties::Execution Slote -> (formutation Unit -> reference (cputoreit): Start Time -> Ame: | End Time => 1000ms;]) in modes (LO) applies to proc.tg_phasel.Rec;<br>Actual Memory Binding => (reference (cpu.internal memory)) applies to proc; | Actual_Processor_Binding => (reference (cpu)) applies to proc;<br>RAMSES_properties::Execution_Slots => ([Computation_Unit => reference (cpu.core2); Start_Time => 800ms; | RAMSES_properties::Execution_Slots => ([Computation_Unit => reference (cpu.core2); Start_Time => 0ms;<br>End_Time => 600ms;1) in modes (LO) applies to proc.tg_phase1.Video; | RAMSES_properties::Execution_Slots => ([Computation_Unit => reference (cpu.core2); Start_Time => 600ms;<br>End_Time => 800ms;]) in modes (LO) applies to proc.tg_phase1.GPS; | <pre>End_iine =&gt; 1100ms; ) in modes (H1), ([Computation_Unit =&gt; reference (cpu.corel); Start_Time =&gt; 300ms; End_Time =&gt; 900ms;]) in modes (L0) applies to proc.tg_bhasel.Nav;</pre> | <pre>run</pre>                                | <pre>properties<br/>RAMSES_properties::Execution_Slots =&gt; ([Computation_Unit =&gt; reference (cpu.corel); Start_Time =&gt; 1100ms;</pre> | <pre>commercial control contro</pre> | <pre>cpu: system cpu.impl {Scheduling_Protocol =&gt; (Static); RAMSES_properties::Is_Processor =&gt; true;}; proc: process proc.impl {Deadline =&gt; 1000ms;};</pre> | system implementation main.impl<br>subcomponents | end main;  | system main | with RANSES properties; | public state of the state of th | package declarative                   | leclarative.aadl ×                          |
|   |   |  |   | <ul> <li>Interview Avoid</li> </ul>   | Memory internal_memory Internal Com  | <ul> <li>C Thread Nav</li> </ul>   | Processor core1   | ► /// Thread Rec                              | <ul> <li>In Thread Video</li> <li>In Thread Log</li> </ul>  | <ul> <li>Interest Group tg_phase1.impt</li> <li>Interest Stab</li> </ul>   | Thread Group tg_phase1   | Process proc                                     | <ul> <li>E System cpu.impl</li> <li>Processor core2</li> </ul> | System cpu  | System main.impl        | System main  | ➡ ➡ Package Public declarative public | PH Outline ×                                |

- 2. The tests also generate a comparison report as a test of the correctness of the de-instantiation. This comparison report is generated and reported within the Console view.
- 3. When de-instantiation takes place, OSATE-DIM also logs information to the console for each performed step.

## Running OSATE-DIM Independently

The user may want to test the de-instantiation of their own custom models. In such a case, the user can use the graphical user interface provided by OSATE-DIM to affect de-instantiation. This interface is described in the Tool Demonstration paper, Mittalet al. "OSATE-DIM Solves the Instance Model-View Update Problem in AADL", provided as extra notes for artifact evaluation.