The SynchAADL2Maude Tool Demo

Kyungmin Bae[1], Peter Ölveczky[2], Abdullah Al-Nayeem[1], and José Meseguer[1]

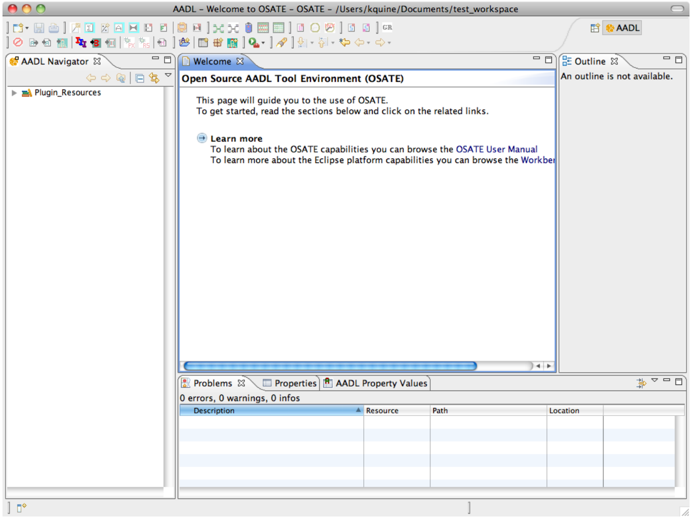
[1]University of Illinois at Urbana-Champaign
[2]University of Oslo

Outline

Outline

OSATE

- OSATE is a toolset for AADL given by a set of Eclipse plugins.
- This is the first screen that you can see when you execute OSATE.



Open Source AADL Tool Environment (OSATE)

This page will guide you to the use of OSATE.
To get started, read the sections below and click on the related links.

Learn more
To learn about the OSATE capabilities you can browse the OSATE User Manual
To learn more about the Eclipse platform capabilities you can browse the Workber

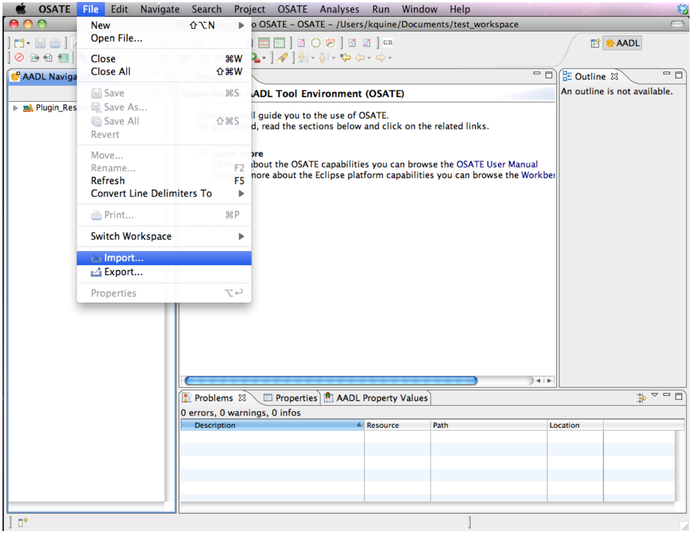
An outline is not available.

0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location

OSATE - Importing an Example (I)

- We start with a simple example.
- First, we will import the Active Standby example.



ADL Tool Environment (OSATE)

ll guide you to the use of OSATE.
d, read the sections below and click on the related links.

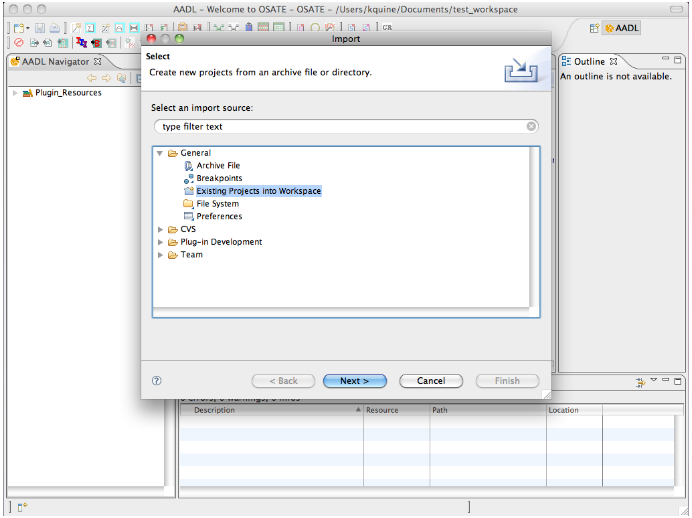
ore
bout the OSATE capabilities you can browse the OSATE User Manual
nore about the Eclipse platform capabilities you can browse the Workbe

An outline is not available.

0 errors, 0 warnings, 0 infos

OSATE - Importing an Example (II)

- The active standby example in out tool webpage can be imported as an existing project.


AADL - Welcome to OSATE - OSATE - /Users/kquine/Documents/test_workspace

Import

Select

Create new projects from an archive file or directory.

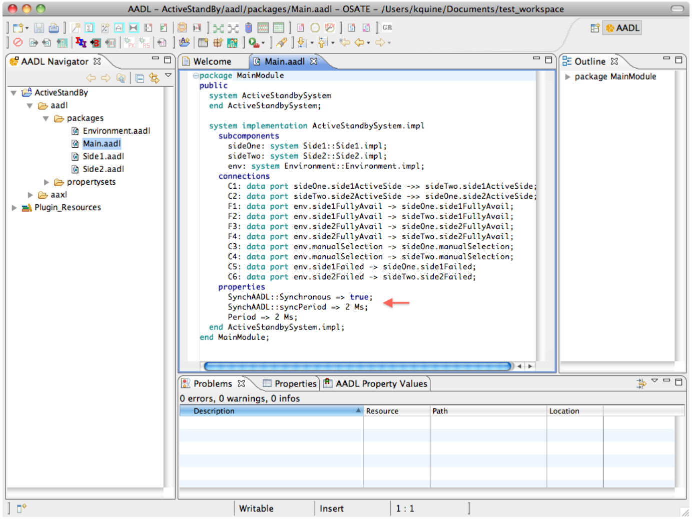
Select an import source:

type filter text

- General
  - Archive File
  - Breakpoints
  - Existing Projects into Workspace
  - File System
  - Preferences
- CVS
- Plug-in Development
- Team

The Active Standby Example - Text

- `Main.aadl` is a top-level system file that shows a brief architecture.
- SynchAADL properties are declared here, to express that this system is in Synchronous AADL

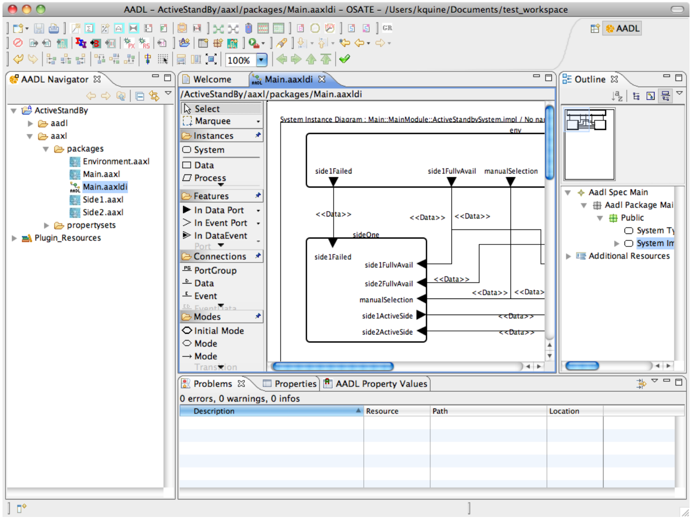


```
package MainModule
public
  system ActiveStandbySystem
  end ActiveStandbySystem;

  system implementation ActiveStandbySystem.impl
    subcomponents
      sideOne: system Side1::Side1.impl;
      sideTwo: system Side2::Side2.impl;
      env: system Environment::Environment.impl;
    connections
      C1: data port sideOne.side1ActiveSide ->> sideTwo.side1ActiveSide;
      C2: data port sideTwo.side2ActiveSide ->> sideOne.side2ActiveSide;
      F1: data port env.side1FullyAvail -> sideOne.side1FullyAvail;
      F2: data port env.side1FullyAvail -> sideTwo.side1FullyAvail;
      F3: data port env.side2FullyAvail -> sideOne.side2FullyAvail;
      F4: data port env.side2FullyAvail -> sideTwo.side2FullyAvail;
      C3: data port env.manualSelection -> sideOne.manualSelection;
      C4: data port env.manualSelection -> sideTwo.manualSelection;
      C5: data port env.side1Failed -> sideOne.side1Failed;
      C6: data port env.side2Failed -> sideTwo.side2Failed;
    properties
      SynchAADL::Synchronous => true;
      SynchAADL::syncPeriod => 2 Ms;
      Period => 2 Ms;
  end ActiveStandbySystem.impl;
end MainModule;
```

0 errors, 0 warnings, 0 infos

The Active Standby Example - Graphic

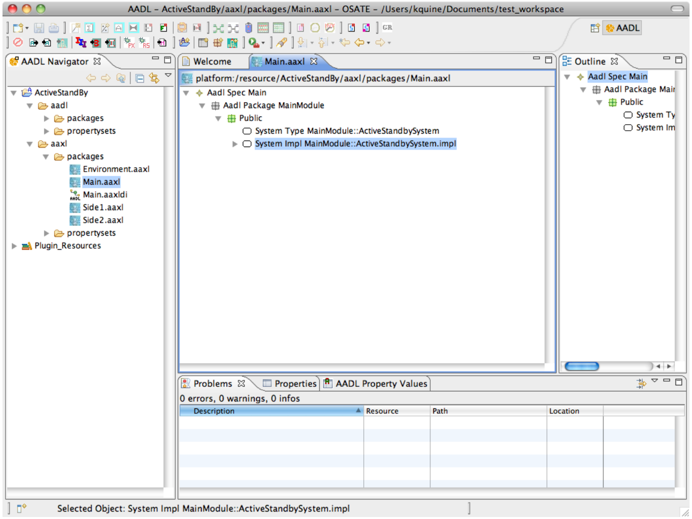
- The AADL graphical model of the active standby example is also given in the file `Main.aaxldi`



0 errors, 0 warnings, 0 infos

The Active Standby Example - XML

- The AADL XML model of the active standby example is automatically generated by OSATE in the file `Main.aaxl`


AADL – ActiveStandBy/aaxl/packages/Main.aaxl – OSATE – /Users/kquine/Documents/test_workspace

platform:/resource/ActiveStandBy/aaxl/packages/Main.aaxl
Aadl Spec Main
Aadl Package MainModule
Public
System Type MainModule::ActiveStandbySystem
System Impl MainModule::ActiveStandbySystem.impl

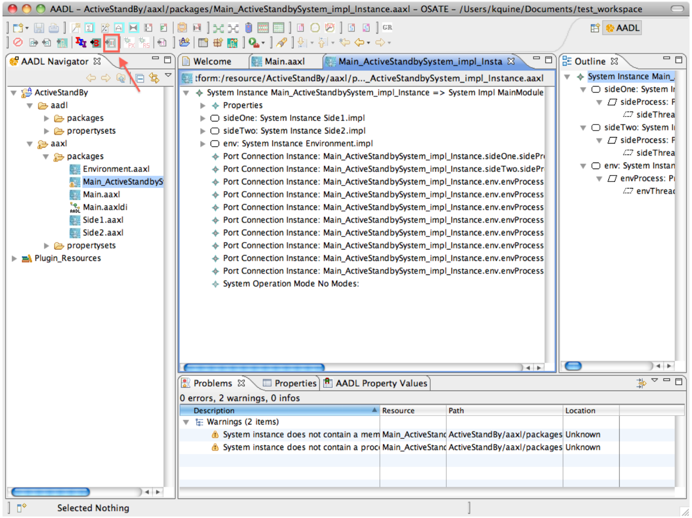
0 errors, 0 warnings, 0 infos

Description	Resource	Path	Location

Selected Object: System Impl MainModule::ActiveStandbySystem.impl

The Active Standby Example - Instance Model

- We can create an instance model from a system implementation by pressing the `Instantiate system` button.
- The top level system implementation of the active standby system is instantiated here.



AADL - ActiveStandBy/aaxl/packages/Main_ActiveStandbySystem_impl_Instance.aaxl - OSATE - /Users/kquine/Documents/test_workspace

```
▼ ActiveStandBy
  ▼ aadl
    ▶ packages
    ▶ propertysets
  ▼ aaxl
    ▼ packages
        Environment.aaxl
        Main_ActiveStandbyS
        Main.aaxl
        Main.aaxldi
        Side1.aaxl
        Side2.aaxl
    ▶ propertysets
▶ Plugin_Resources
```

```
:form:/resource/ActiveStandBy/aaxl/p..._ActiveStandbySystem_impl_Instance.aaxl
▼ System Instance Main_ActiveStandbySystem_impl_Instance => System Impl MainModule
  ▶ Properties
  ▶ sideOne: System Instance Side1.impl
  ▶ sideTwo: System Instance Side2.impl
  ▶ env: System Instance Environment.impl
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.sideOne.sidePr
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.sideTwo.sidePr
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    Port Connection Instance: Main_ActiveStandbySystem_impl_Instance.env.envProcess
    System Operation Mode No Modes:
```

```
▼ System Instance Main_
  ▼ sideOne: System In
    ▼ sideProcess: F
        sideThrea
  ▼ sideTwo: System In
    ▼ sideProcess: F
        sideThrea
  ▼ env: System Instan
    ▼ envProcess: Pr
        envThread
```

0 errors, 2 warnings, 0 infos

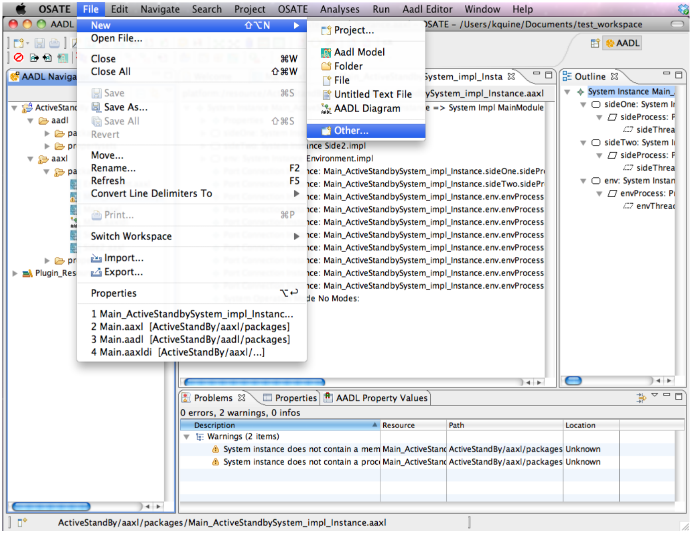
Description	Resource	Path	Location
▼ Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

Selected Nothing

Outline

Invoking the SynchAADL2Maude Window

- The SynchAADL2-Maude window can be invoked from an AADL instant model.



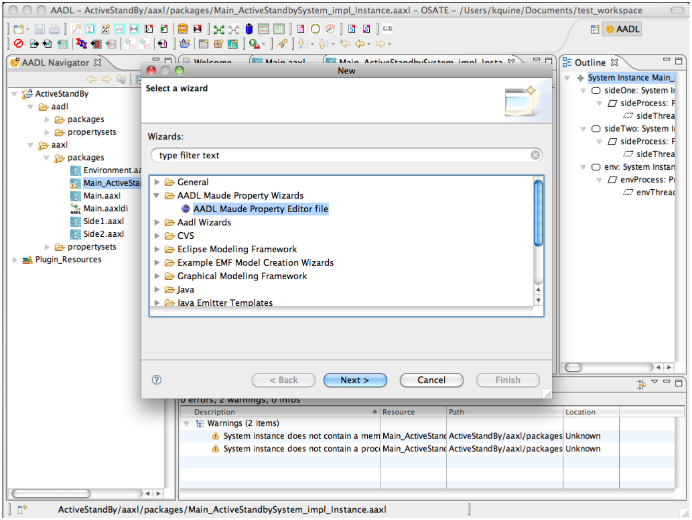
0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

ActiveStandBy/aaxl/packages/Main_ActiveStandbySystem_impl_Instance.aaxl

AADL Maude Property Editor Wizards

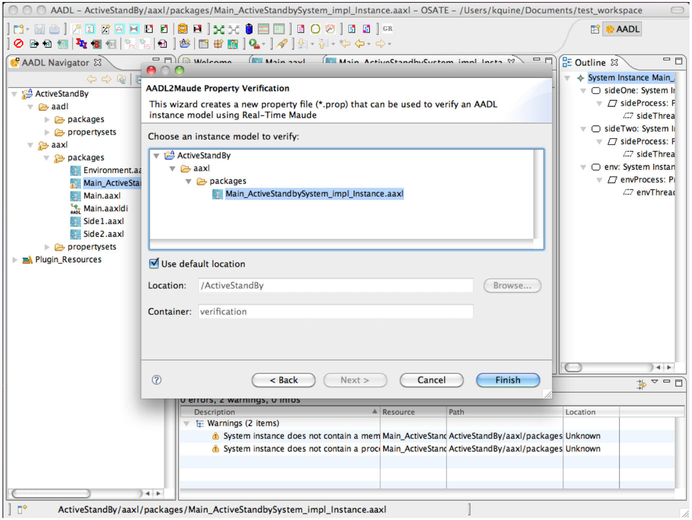
- From the `File` menu, we can create an AADL Maude Property Editor file.



Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

Creating an AADL Maude Property File

- We can choose any valid AADL instance model from the wizard.



AADL – ActiveStandBy/aaxl/packages/Main_ActiveStandbySystem_impl_Instance.aaxl – OSATE – /Users/kquine/Documents/test_workspace

AADL Navigator

- ActiveStandBy
  - aadl
    - packages
    - propertysets
  - aaxl
    - packages
      - Environment.aa
      - Main_ActiveSta
      - Main.aaxl
      - Main.aaxldi
      - Side1.aaxl
      - Side2.aaxl
    - propertysets
- Plugin_Resources

AADL2Maude Property Verification

This wizard creates a new property file (*.prop) that can be used to verify an AADL instance model using Real-Time Maude

Choose an instance model to verify:

- ActiveStandBy
  - aaxl
    - packages
      - Main_ActiveStandbySystem_impl_Instance.aaxl

☑ Use default location

Location: /ActiveStandBy

Container: verification

< Back | Next > | Cancel | Finish

Outline

- System Instance Main_
  - sideOne: System I
    - sideProcess: F
      - sideThrea
  - sideTwo: System I
    - sideProcess: F
      - sideThrea
  - env: System Instan
    - envProcess: P
      - envThrea

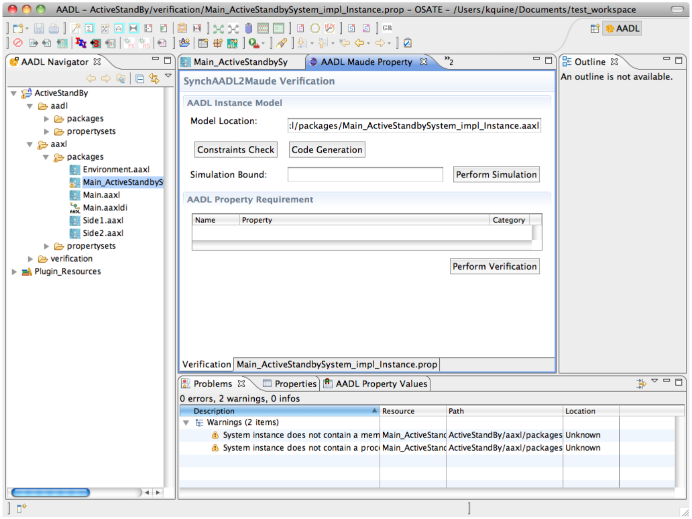
0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

ActiveStandBy/aaxl/packages/Main_ActiveStandbySystem_impl_Instance.aaxl

The SynchAADL2Maude Window

- This screen shows the SynchAADL2-Maude window.
- There are four buttons in this window: `Constraints Check`, `Code Generation`, `Perform Simulation`, and `Perform Verification`.



AADL – ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop – OSATE – /Users/kquine/Documents/test_workspace

An outline is not available.

SynchAADL2Maude Verification

AADL Instance Model

Model Location: :l/packages/Main_ActiveStandbySystem_impl_Instance.aaxl

Simulation Bound:

AADL Property Requirement

Name	Property	Category

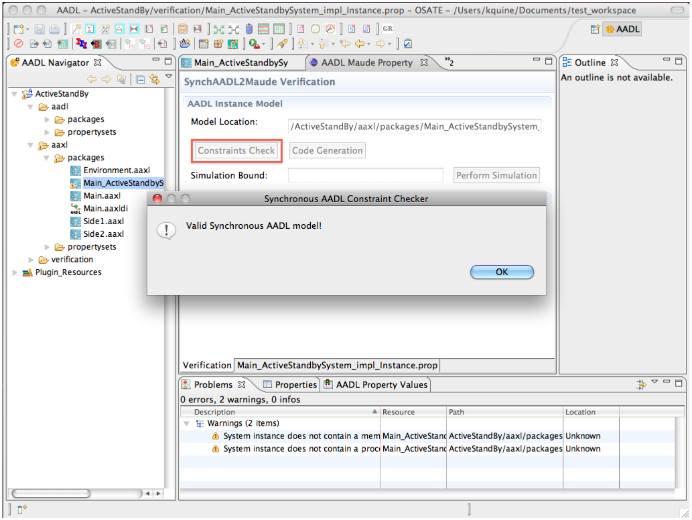
0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

Outline

Checking SynchAADL Constraints

- We can check SynchAADL constraints by clicking on the Constraints Check button.



AADL – ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop – OSATE – /Users/kquine/Documents/test_workspace

SynchAADL2Maude Verification

AADL Instance Model

Model Location: /ActiveStandBy/aaxl/packages/Main_ActiveStandbySystem_

Simulation Bound:

An outline is not available.

Synchronous AADL Constraint Checker

Valid Synchronous AADL model!

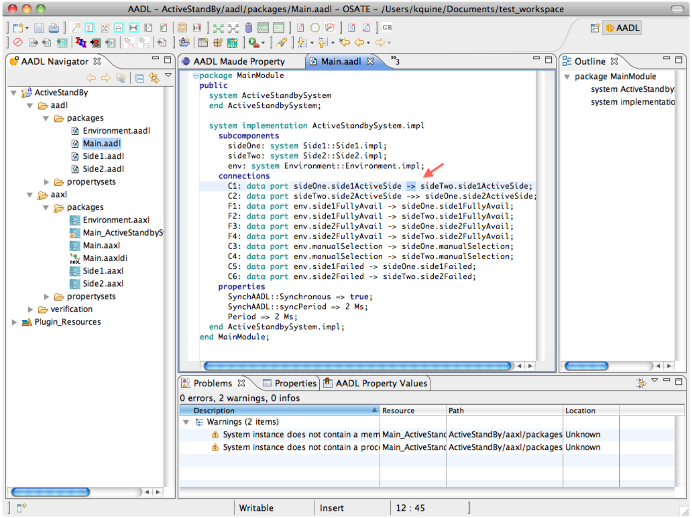
Main_ActiveStandbySystem_impl_Instance.prop

0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

SynchAADL Constraints - Erroneous Cases (I)

- What if some SynchAADL constraint is not satisfied?
- We add an invalid immediate connection, and see what happened.



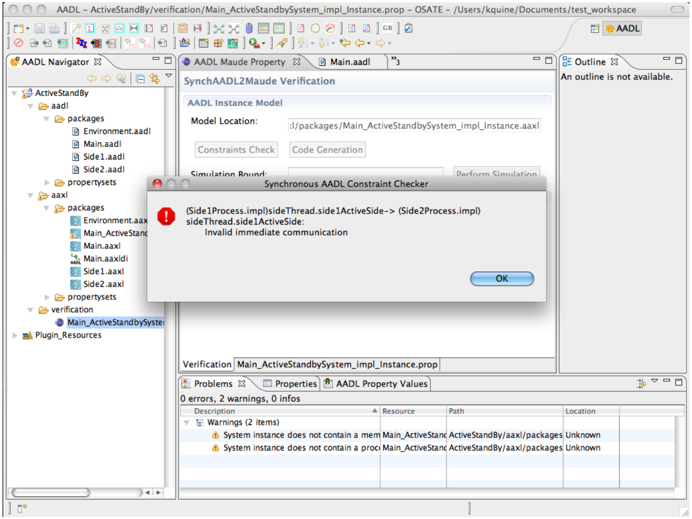
```
package MainModule
public
  system ActiveStandbySystem
  end ActiveStandbySystem;

  system implementation ActiveStandbySystem.impl
    subcomponents
      sideOne: system Side1::Side1.impl;
      sideTwo: system Side2::Side2.impl;
      env: system Environment::Environment.impl;
    connections
      C1: data port sideOne.side1ActiveSide -> sideTwo.side1ActiveSide;
      C2: data port sideTwo.side2ActiveSide ->> sideOne.side2ActiveSide;
      F1: data port env.side1FullyAvail -> sideOne.side1FullyAvail;
      F2: data port env.side1FullyAvail -> sideTwo.side1FullyAvail;
      F3: data port env.side2FullyAvail -> sideOne.side2FullyAvail;
      F4: data port env.side2FullyAvail -> sideTwo.side2FullyAvail;
      C3: data port env.manualSelection -> sideOne.manualSelection;
      C4: data port env.manualSelection -> sideTwo.manualSelection;
      C5: data port env.side1Failed -> sideOne.side1Failed;
      C6: data port env.side2Failed -> sideTwo.side2Failed;
    properties
      SynchAADL::Synchronous => true;
      SynchAADL::syncPeriod => 2 Ms;
      Period => 2 Ms;
  end ActiveStandbySystem.impl;
end MainModule;
```

0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

SynchAADL Constraints - Erroneous Cases (II)

- Our tool then notifies errors.



Synchronous AADL Constraint Checker

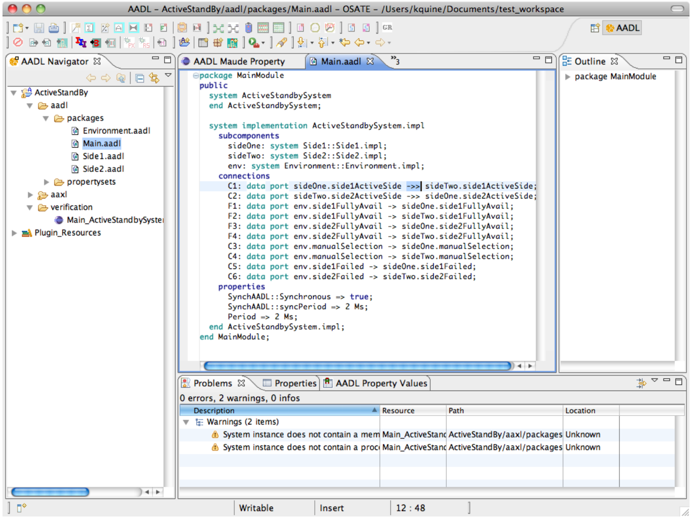
```
(Side1Process.impl)sideThread.side1ActiveSide-> (Side2Process.impl)
sideThread.side1ActiveSide:
        Invalid immediate communication
```

```
0 errors, 2 warnings, 0 infos
Description                              Resource          Path                          Location
Warnings (2 items)
  System instance does not contain a mem  Main_ActiveStanc  ActiveStandBy/aaxl/packages   Unknown
  System instance does not contain a proc Main_ActiveStanc  ActiveStandBy/aaxl/packages   Unknown
```

Outline

The Active Standby Example

- Let us go back to the correct model.



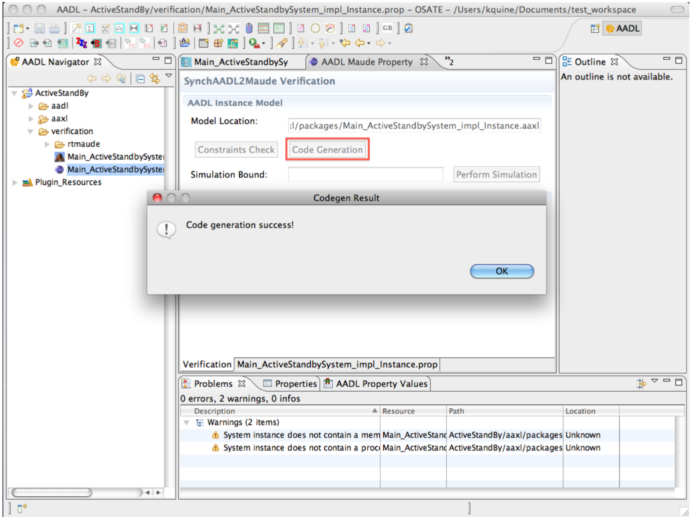
```
package MainModule
public
  system ActiveStandbySystem
  end ActiveStandbySystem;

  system implementation ActiveStandbySystem.impl
    subcomponents
      sideOne: system Side1::Side1.impl;
      sideTwo: system Side2::Side2.impl;
      env: system Environment::Environment.impl;
    connections
      C1: data port sideOne.side1ActiveSide ->> sideTwo.side1ActiveSide;
      C2: data port sideTwo.side2ActiveSide ->> sideOne.side2ActiveSide;
      F1: data port env.side1FullyAvail -> sideOne.side1FullyAvail;
      F2: data port env.side1FullyAvail -> sideTwo.side1FullyAvail;
      F3: data port env.side2FullyAvail -> sideOne.side2FullyAvail;
      F4: data port env.side2FullyAvail -> sideTwo.side2FullyAvail;
      C3: data port env.manualSelection -> sideOne.manualSelection;
      C4: data port env.manualSelection -> sideTwo.manualSelection;
      C5: data port env.side1Failed -> sideOne.side1Failed;
      C6: data port env.side2Failed -> sideTwo.side2Failed;
    properties
      SynchAADL::Synchronous => true;
      SynchAADL::syncPeriod => 2 Ms;
      Period => 2 Ms;
  end ActiveStandbySystem.impl;
end MainModule;
```

0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

Real-Time Maude Code Generation (I)

- We can automatically create the corresponding Real-Time Maude model from a Synchronous AADL model by clicking on the `Code Generation` button.


AADL - ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop - OSATE - /Users/kquine/Documents/test_workspace

SynchAADL2Maude Verification

AADL Instance Model

Model Location: :l/packages/Main_ActiveStandbySystem_impl_Instance.aaxl

Simulation Bound:

Codegen Result

Code generation success!

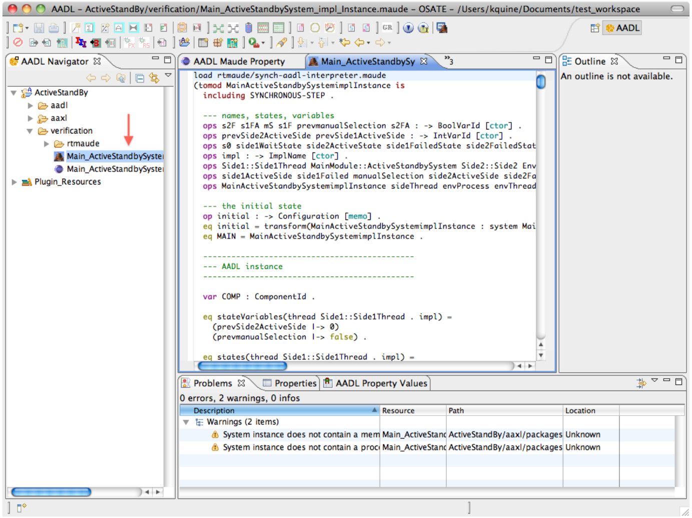
Outline: An outline is not available.

0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

Real-Time Maude Code Generation (II)

- We can find the generated Real-Time Maude model on the AADL navigator sidebar.



AADL - ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.maude - OSATE - /Users/kquine/Documents/test_workspace

```
load rtmaude/synch-aadl-interpreter.maude
(tomod MainActiveStandbySystemimplInstance is
  including SYNCHRONOUS-STEP .

  --- names, states, variables
  ops s2F s1FA mS s1F prevmanualSelection s2FA : -> BoolVarId [ctor] .
  ops prevSide2ActiveSide prevSide1ActiveSide : -> IntVarId [ctor] .
  ops s0 side1WaitState side2ActiveState side1FailedState side2FailedStat
  ops impl : -> ImplName [ctor] .
  ops Side1::Side1Thread MainModule::ActiveStandbySystem Side2::Side2 Env
  ops side1ActiveSide side1Failed manualSelection side2ActiveSide side2Fa
  ops MainActiveStandbySystemimplInstance sideThread envProcess envThread

  --- the initial state
  op initial : -> Configuration [memo] .
  eq initial = transform(MainActiveStandbySystemimplInstance : system Mai
  eq MAIN = MainActiveStandbySystemimplInstance .

  -------------------------------------------
  --- AADL instance
  -------------------------------------------

  var COMP : ComponentId .

  eq stateVariables(thread Side1::Side1Thread . impl) =
    (prevSide2ActiveSide |-> 0)
    (prevmanualSelection |-> false) .

  eq states(thread Side1::Side1Thread . impl) =
```

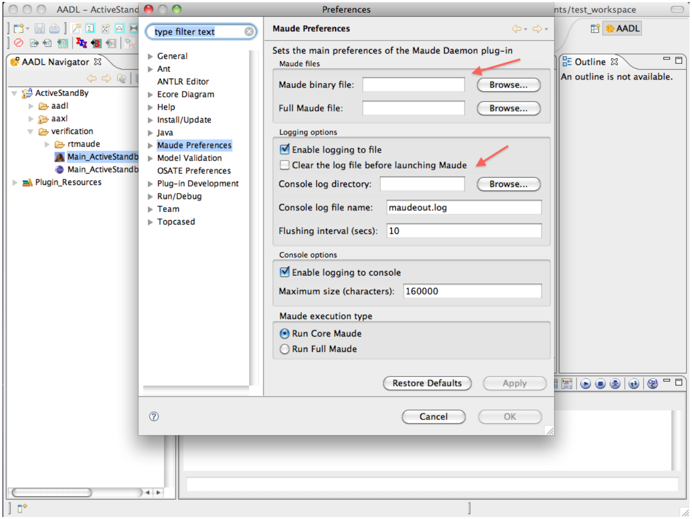
An outline is not available.

0 errors, 2 warnings, 0 infos

Description	Resource	Path	Location
Warnings (2 items)			
System instance does not contain a mem	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown
System instance does not contain a proc	Main_ActiveStanc	ActiveStandBy/aaxl/packages	Unknown

Maude Development Tool Setting

- When a Maude file is first executed, the MDT setting window is popped-up.
- The correct paths of both a Maude binary file and a Full Maude file should be inserted.
- If "logging to file" is enabled, we should also insert a console log directory.



Maude Preferences

Sets the main preferences of the Maude Daemon plug-in

Maude files

Maude binary file: [ ] Browse...

Full Maude file: [ ] Browse...

Logging options

[✓] Enable logging to file

[ ] Clear the log file before launching Maude

Console log directory: [ ] Browse...

Console log file name: maudeout.log

Flushing interval (secs): 10

Console options

[✓] Enable logging to console

Maximum size (characters): 160000

Maude execution type

[✓ Run Core Maude]

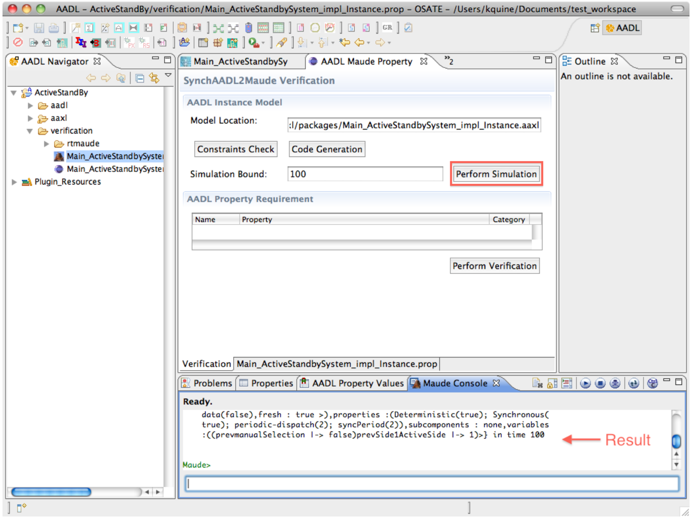
Run Full Maude

Restore Defaults Apply

Cancel OK

SynchAADL Simulation in Real-Time Maude

- We can simulate a given model within some bound by pressing the `Perform Simulation` button.
- The result will be shown in the `Maude Console`.


AADL - ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop - OSATE - /Users/kquine/Documents/test_workspace

SynchAADL2Maude Verification

AADL Instance Model

Model Location: :l/packages/Main_ActiveStandbySystem_impl_Instance.aaxl

Simulation Bound: 100

AADL Property Requirement

Name	Property	Category

An outline is not available.

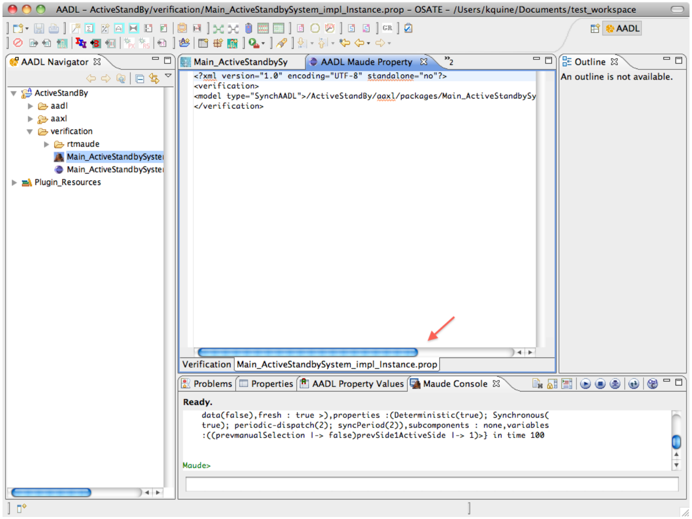
Ready.

```
    data(false),fresh : true >),properties :(Deterministic(true); Synchronous(
    true); periodic-dispatch(2); syncPeriod(2)),subcomponents : none,variables
    :((prevmanualSelection |-> false)prevSide1ActiveSide |-> 1)>} in time 100

Maude>
```

Outline

XML Property File (I)

- AADL Maude property files are actually XML files.
- We can see and modify the content of the file by clicking on the right tab at the bottom.

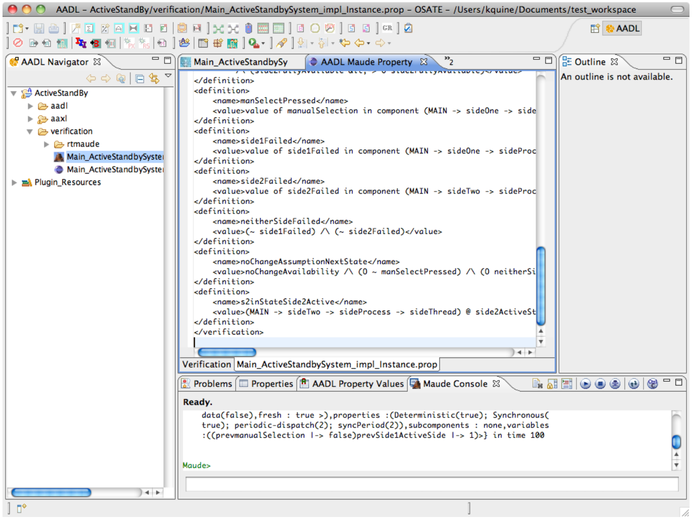


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<verification>
<model type="SynchAADL">/ActiveStandBy/aaxl/packages/Main_ActiveStandbySy
</verification>
```

```
Ready.
    data(false),fresh : true >),properties :(Deterministic(true); Synchronous(
    true); periodic-dispatch(2); syncPeriod(2)),subcomponents : none,variables
    :((prevmanualSelection |-> false)prevSide1ActiveSide |-> 1)>} in time 100

Maude>
```

XML Property File (II)

- The LTL formulas can be defined by `definition` tags.
- The LTL specifications to be verified are defined in `command` tags.
- Let us copy and paste the property definitions from the active standby example in the tool webpage.

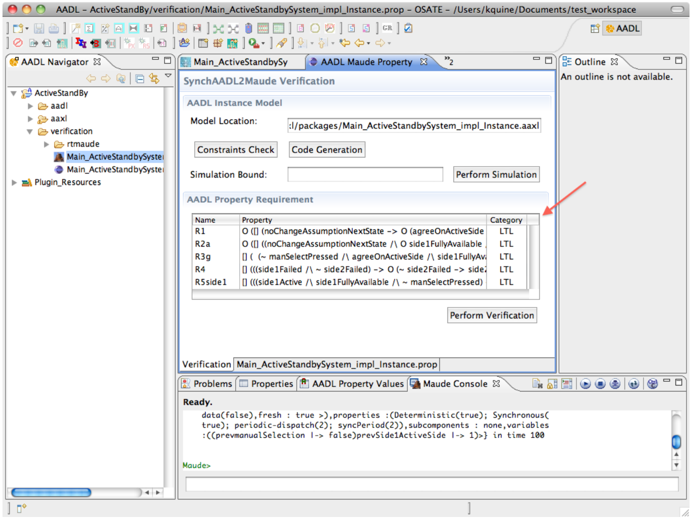


```
</definition>
<definition>
    <name>manSelectPressed</name>
    <value>value of manualSelection in component (MAIN -> sideOne -> side
</definition>
<definition>
    <name>side1Failed</name>
    <value>value of side1Failed in component (MAIN -> sideOne -> sideProc
</definition>
<definition>
    <name>side2Failed</name>
    <value>value of side2Failed in component (MAIN -> sideTwo -> sideProc
</definition>
<definition>
    <name>neitherSideFailed</name>
    <value>(~ side1Failed) /\ (~ side2Failed)</value>
</definition>
<definition>
    <name>noChangeAssumptionNextState</name>
    <value>noChangeAvailability /\ (O ~ manSelectPressed) /\ (O neitherSi
</definition>
<definition>
    <name>s2inStateSide2Active</name>
    <value>(MAIN -> sideTwo -> sideProcess -> sideThread) @ side2ActiveSt
</definition>
</verification>
```

```
Ready.
    data(false),fresh : true >),properties :(Deterministic(true); Synchronous(
    true); periodic-dispatch(2); syncPeriod(2)),subcomponents : none,variables
    :((prevmanualSelection |-> false)prevSide1ActiveSide |-> 1)>} in time 100

Maude>
```

Model Checking LTL Specifications (I)

- The LTL specification to be verified are shown in the `AADL Property Requirement` table.



AADL - ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop - OSATE - /Users/kquine/Documents/test_workspace

SynchAADL2Maude Verification

AADL Instance Model

Model Location: :l/packages/Main_ActiveStandbySystem_impl_Instance.aaxl

Constraints Check | Code Generation

Simulation Bound: | Perform Simulation

AADL Property Requirement

Name	Property	Category
R1	O ([] (noChangeAssumptionNextState -> O (agreeOnActiveSide	LTL
R2a	O ([] ((noChangeAssumptionNextState /\ O side1FullyAvailable	LTL
R3g	[] ((~ manSelectPressed /\ agreeOnActiveSide /\ side1FullyAv	LTL
R4	[] (((side1Failed /\ ~ side2Failed) -> O (~ side2Failed -> side2	LTL
R5side1	[] (((side1Active /\ side1FullyAvailable /\ ~ manSelectPressed)	LTL

Perform Verification

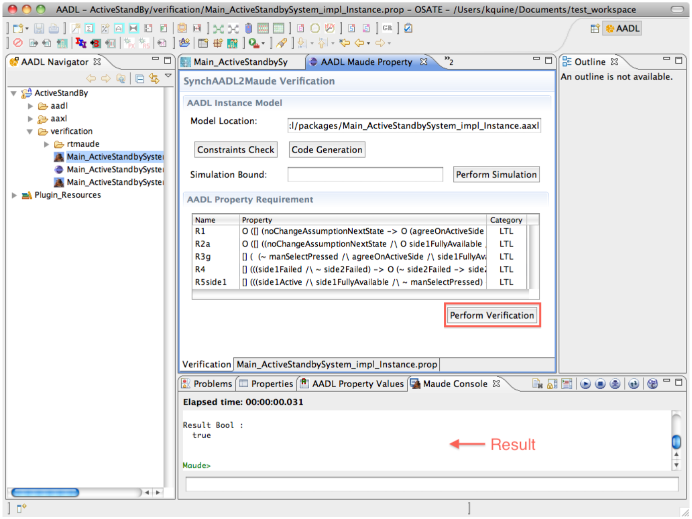
Verification | Main_ActiveStandbySystem_impl_Instance.prop

Ready.

```
    data(false),fresh : true >),properties :(Deterministic(true); Synchronous(
    true); periodic-dispatch(2); syncPeriod(2)),subcomponents : none,variables
    :((prevmanualSelection |-> false)prevSide1ActiveSide |-> 1)>} in time 100


Maude>
```

Model Checking LTL Specifications (II)

- When we press the `Perform Verification` button, the LTL properties in the table are model checked in Real-Time Maude.
- The model checking result will be shown in the `Maude Console`.



AADL - ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop - OSATE - /Users/kquine/Documents/test_workspace

SynchAADL2Maude Verification

AADL Instance Model

Model Location: :l/packages/Main_ActiveStandbySystem_impl_Instance.aaxl

Simulation Bound:

AADL Property Requirement

Name	Property	Category
R1	O ([] (noChangeAssumptionNextState -> O (agreeOnActiveSide	LTL
R2a	O ([] ((noChangeAssumptionNextState /\ O side1FullyAvailable	LTL
R3g	[] ((~ manSelectPressed /\ agreeOnActiveSide /\ side1FullyAv	LTL
R4	[] (((side1Failed /\ ~ side2Failed) -> O (~ side2Failed -> side	LTL
R5side1	[] (((side1Active /\ side1FullyAvailable /\ ~ manSelectPressed)	LTL

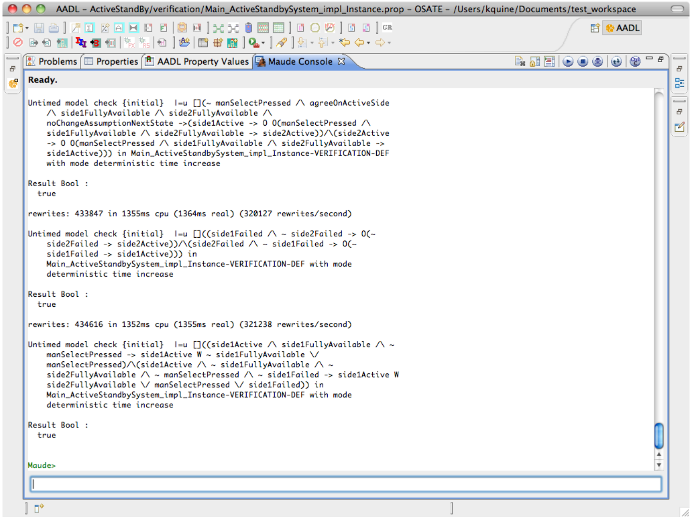
An outline is not available.

Elapsed time: 00:00:00.031

```
Result Bool :
  true

Maude>
```

Model Checking LTL Specifications (III)

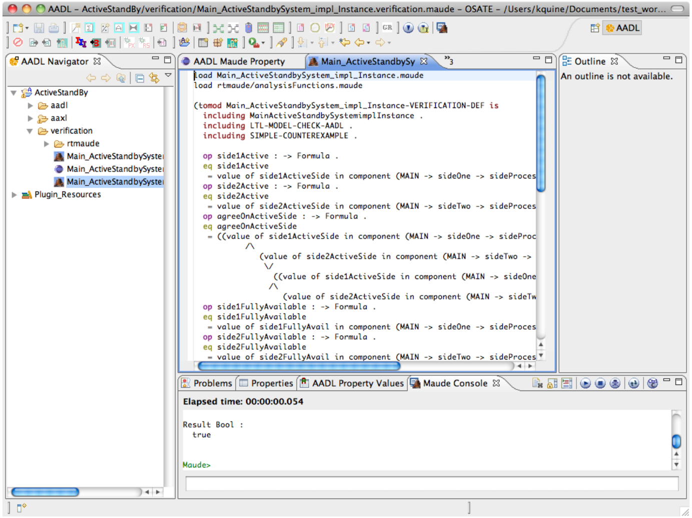
- Here is the model checking result of the active standby example in a larger window.


AADL – ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop – OSATE – /Users/kquine/Documents/test_workspace
AADL
Problems
Properties
AADL Property Values
Maude Console
Ready.

```
Untimed model check {initial}  |=u [](~ manSelectPressed /\ agreeOnActiveSide
    /\ side1FullyAvailable /\ side2FullyAvailable /\
    noChangeAssumptionNextState ->(side1Active -> O O(manSelectPressed /\
    side1FullyAvailable /\ side2FullyAvailable -> side2Active))/\(side2Active
    -> O O(manSelectPressed /\ side1FullyAvailable /\ side2FullyAvailable ->
    side1Active))) in Main_ActiveStandbySystem_impl_Instance-VERIFICATION-DEF
    with mode deterministic time increase

Result Bool :
  true

rewrites: 433847 in 1355ms cpu (1364ms real) (320127 rewrites/second)

Untimed model check {initial}  |=u []((side1Failed /\ ~ side2Failed -> O(~
    side2Failed -> side2Active))/\(side2Failed /\ ~ side1Failed -> O(~
    side1Failed -> side1Active))) in
    Main_ActiveStandbySystem_impl_Instance-VERIFICATION-DEF with mode
    deterministic time increase

Result Bool :
  true

rewrites: 434616 in 1352ms cpu (1355ms real) (321238 rewrites/second)

Untimed model check {initial}  |=u []((side1Active /\ side1FullyAvailable /\ ~
    manSelectPressed -> side1Active W ~ side1FullyAvailable \/
    manSelectPressed)/\(side1Active /\ ~ side1FullyAvailable /\ ~
    side2FullyAvailable /\ ~ manSelectPressed /\ ~ side1Failed -> side1Active W
    side2FullyAvailable \/ manSelectPressed \/ side1Failed)) in
    Main_ActiveStandbySystem_impl_Instance-VERIFICATION-DEF with mode
    deterministic time increase

Result Bool :
  true

Maude>
```

Model Checking LTL Specifications (IV)

- SynchAADL2-Maude creates the Real-Time Maude verification model from a XML property file.
- The verification model can be also found in the AADL Navigator sidebar.

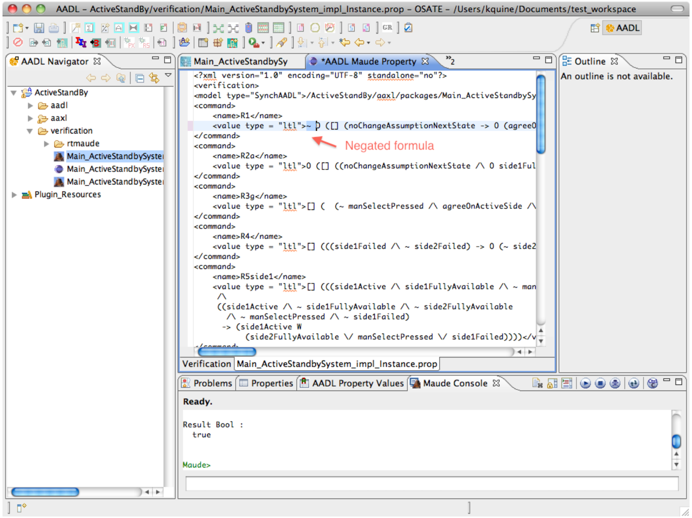


```
load Main_ActiveStandbySystem_impl_Instance.maude
load rtmaude/analysisFunctions.maude

(tomod Main_ActiveStandbySystem_impl_Instance-VERIFICATION-DEF is
  including MainActiveStandbySystemimplInstance .
  including LTL-MODEL-CHECK-AADL .
  including SIMPLE-COUNTEREXAMPLE .

  op side1Active : -> Formula .
  eq side1Active
   = value of side1ActiveSide in component (MAIN -> sideOne -> sideProces
  op side2Active : -> Formula .
  eq side2Active
   = value of side2ActiveSide in component (MAIN -> sideTwo -> sideProces
  op agreeOnActiveSide : -> Formula .
  eq agreeOnActiveSide
   = ((value of side1ActiveSide in component (MAIN -> sideOne -> sideProc
            /\
              (value of side2ActiveSide in component (MAIN -> sideTwo ->
               \/
                ((value of side1ActiveSide in component (MAIN -> sideOne
                /\
                  (value of side2ActiveSide in component (MAIN -> sideTw
  op side1FullyAvailable : -> Formula .
  eq side1FullyAvailable
   = value of side1FullyAvail in component (MAIN -> sideOne -> sideProces
  op side2FullyAvailable : -> Formula .
  eq side2FullyAvailable
   = value of side2FullyAvail in component (MAIN -> sideTwo -> sideProces
```

```
Elapsed time: 00:00:00.054

Result Bool :
  true

Maude>
```

Counterexamples (I)

- If a given LTL property is not satisfied in a model, then a counterexample is generated.
- We illustrate such counterexamples with an incorrect LTL specification for the active standby model.

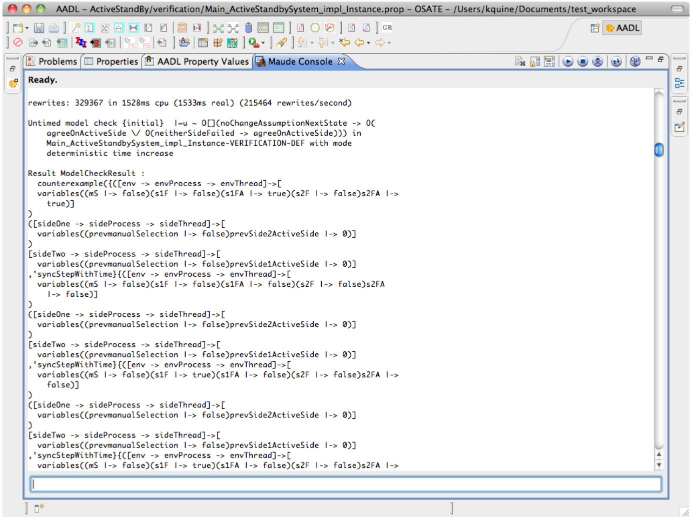


```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<verification>
<model type="SynchAADL">/ActiveStandBy/aaxl/packages/Main_ActiveStandbySy
<command>
    <name>R1</name>
    <value type = "ltl">~ O ([] (noChangeAssumptionNextState -> O (agreeO
</command>
<command>
    <name>R2a</name>
    <value type = "ltl">O ([] ((noChangeAssumptionNextState /\ O side1Ful
</command>
<command>
    <name>R3g</name>
    <value type = "ltl">[] (  (~ manSelectPressed /\ agreeOnActiveSide /\
</command>
<command>
    <name>R4</name>
    <value type = "ltl">[] (((side1Failed /\ ~ side2Failed) -> O (~ side2
</command>
<command>
    <name>R5side1</name>
    <value type = "ltl">[] (((side1Active /\ side1FullyAvailable /\ ~ man
     /\
     ((side1Active /\ ~ side1FullyAvailable /\ ~ side2FullyAvailable
       /\ ~ manSelectPressed /\ ~ side1Failed)
      -> (side1Active W
          (side2FullyAvailable \/ manSelectPressed \/ side1Failed)))</v
```

```
Ready.

Result Bool :
  true

Maude>
```

Counterexamples (II)

- Here is a generated counterexample in SynchAADL2-Maude.
- For each state, a component name and its local variables are displayed.


AADL - ActiveStandBy/verification/Main_ActiveStandbySystem_impl_Instance.prop - OSATE - /Users/kquine/Documents/test_workspace
AADL
Problems
Properties
AADL Property Values
Maude Console
Ready.

```
rewrites: 329367 in 1528ms cpu (1533ms real) (215464 rewrites/second)

Untimed model check {initial}  |=u ~ O[](noChangeAssumptionNextState -> O(
    agreeOnActiveSide \/ O(neitherSideFailed -> agreeOnActiveSide))) in
    Main_ActiveStandbySystem_impl_Instance-VERIFICATION-DEF with mode
    deterministic time increase

Result ModelCheckResult :
  counterexample({([env -> envProcess -> envThread]->[
  variables((mS |-> false)(s1F |-> false)(s1FA |-> true)(s2F |-> false)s2FA |->
    true)]
)
([sideOne -> sideProcess -> sideThread]->[
  variables((prevmanualSelection |-> false)prevSide2ActiveSide |-> 0)]
)
[sideTwo -> sideProcess -> sideThread]->[
  variables((prevmanualSelection |-> false)prevSide1ActiveSide |-> 0)]
,'syncStepWithTime}{([env -> envProcess -> envThread]->[
  variables((mS |-> false)(s1F |-> false)(s1FA |-> false)(s2F |-> false)s2FA
    |-> false)]
)
([sideOne -> sideProcess -> sideThread]->[
  variables((prevmanualSelection |-> false)prevSide2ActiveSide |-> 0)]
)
[sideTwo -> sideProcess -> sideThread]->[
  variables((prevmanualSelection |-> false)prevSide1ActiveSide |-> 0)]
,'syncStepWithTime}{([env -> envProcess -> envThread]->[
  variables((mS |-> false)(s1F |-> true)(s1FA |-> false)(s2F |-> false)s2FA |->
    false)]
)
([sideOne -> sideProcess -> sideThread]->[
  variables((prevmanualSelection |-> false)prevSide2ActiveSide |-> 0)]
)
[sideTwo -> sideProcess -> sideThread]->[
  variables((prevmanualSelection |-> false)prevSide1ActiveSide |-> 0)]
,'syncStepWithTime}{([env -> envProcess -> envThread]->[
  variables((mS |-> false)(s1F |-> true)(s1FA |-> false)(s2F |-> false)s2FA |->
```

Thank you!