HybridSynchAADL

Tutorial

Outline

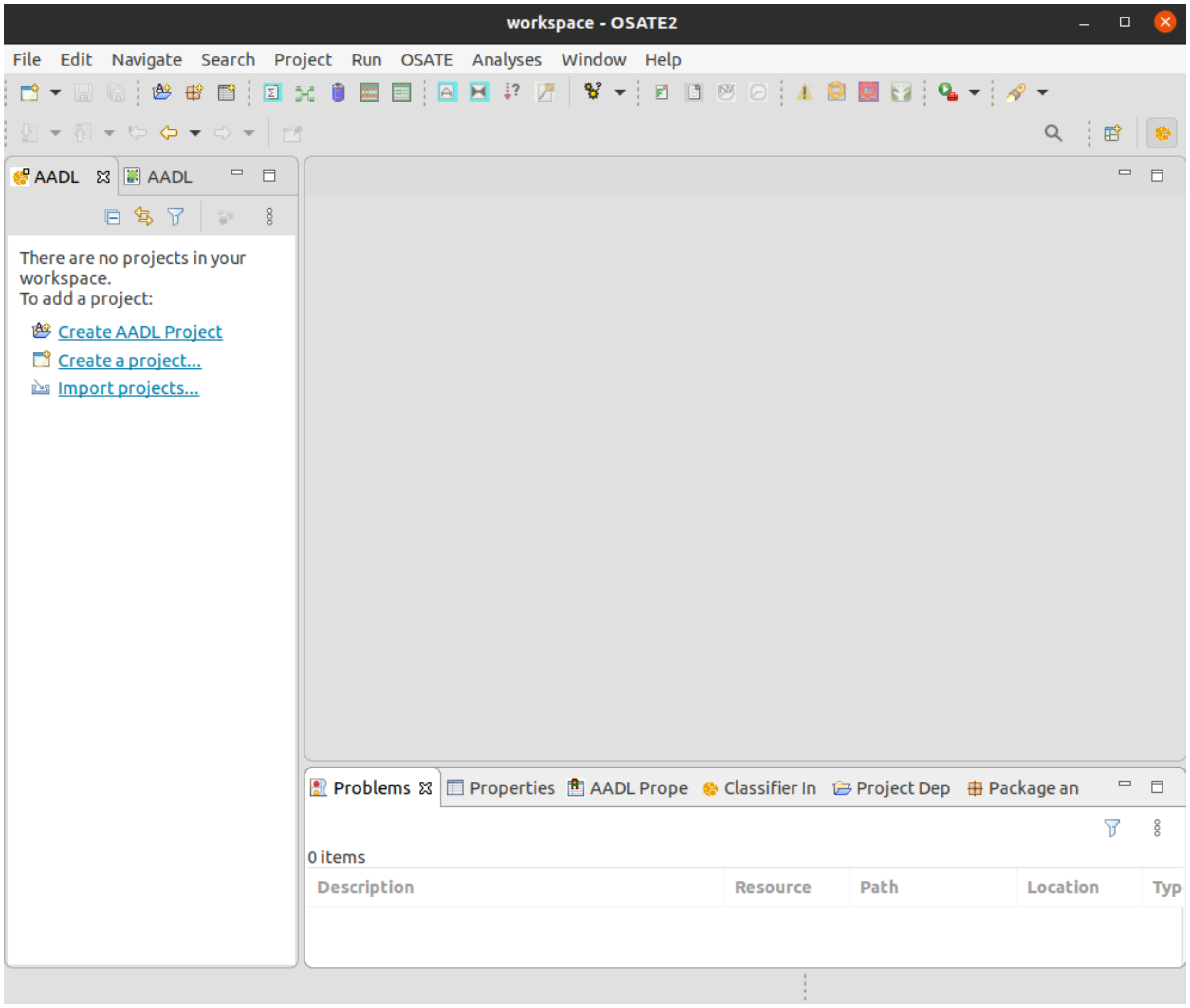
1. Basic OSATE
2. Creating Property Specification Files (PSPC)
3. HybridSynchAADL Constraints Checker
4. Maude Code Generation
5. Formal Analysis

Outline

1. Basic OSATE
2. Creating Property Specification Files (PSPC)
3. HybridSynchAADL Constraints Checker
4. Maude Code Generation
5. Formal Analysis

Running OSATE

- You will see this window when you execute OSATE.

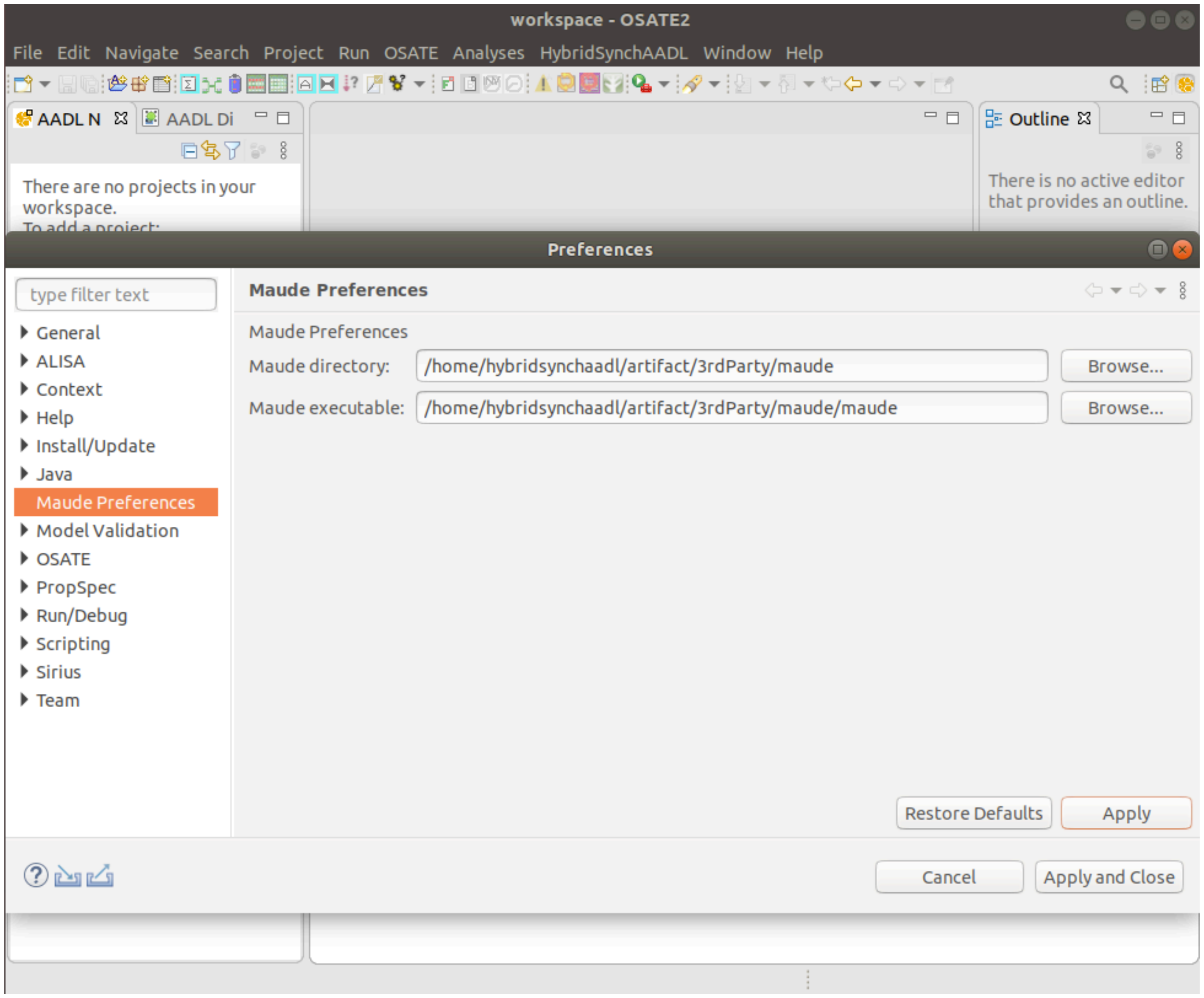


There are no projects in your workspace.
To add a project:

0 items

Maude Preferences

- Before importing an example project, set the proper Maude preferences.
- Open Preferences menu by clicking Menu → Window → Preferences.
- Set Maude directory and executable file location.



There are no projects in your workspace.
To add a project:

There is no active editor that provides an outline.

Maude Preferences

Maude Preferences

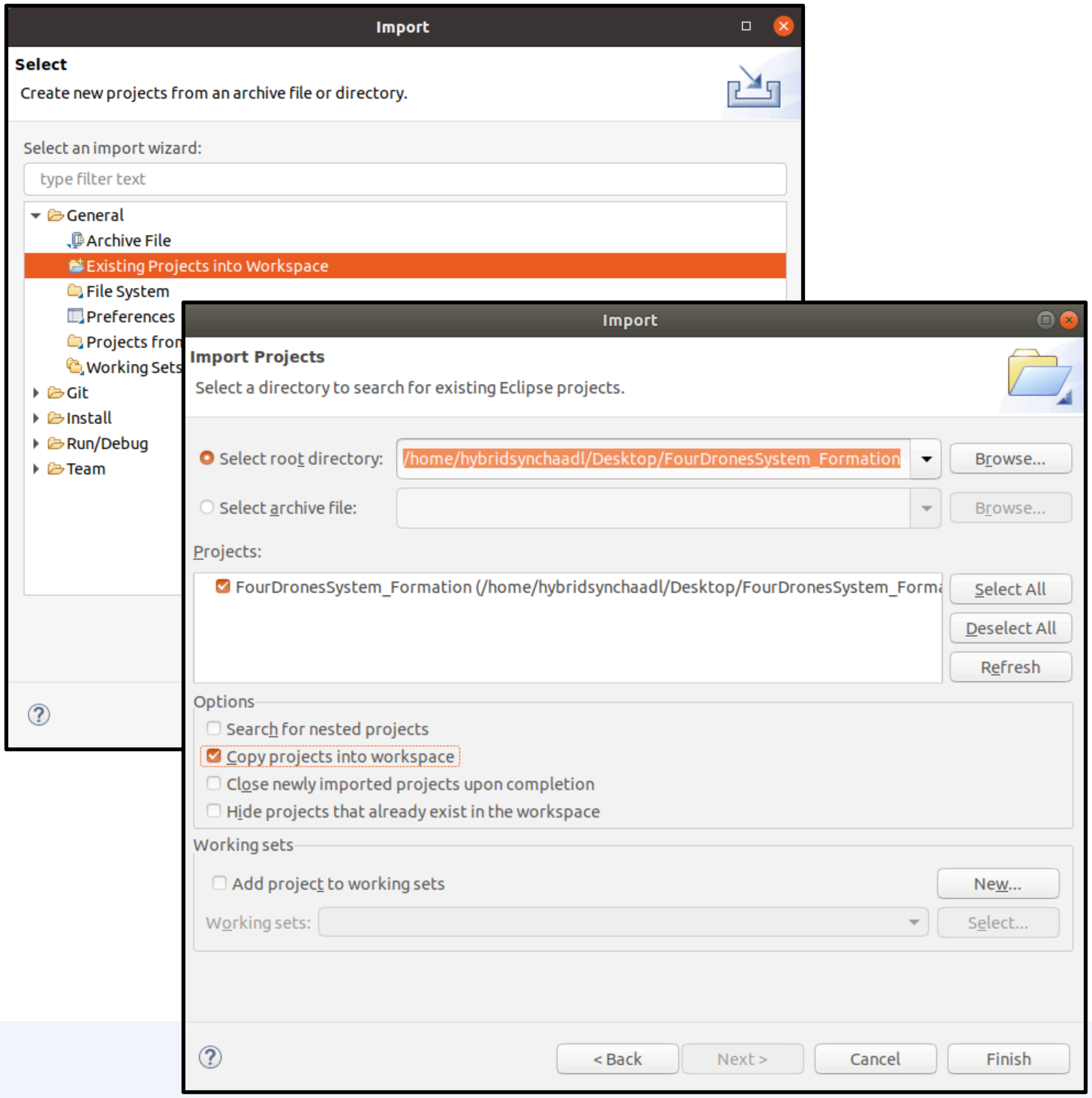
Maude directory: /home/hybridsynchaadl/artifact/3rdParty/maude

Maude executable: /home/hybridsynchaadl/artifact/3rdParty/maude/maude

OSATE - Importing an Example

- We start with a simple example, namely, `FourDronesSystem_Formation` in the directory `models/hybridsynchaadl`.

- To import the example, choose
  - Menu → File → Import → General → Existing Projects into Workspace.



Import

Select

Create new projects from an archive file or directory.

Select an import wizard:

type filter text

Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: /home/hybridsynchaadl/Desktop/FourDronesSystem_Formation

○ Select archive file:

Projects:

☑ FourDronesSystem_Formation (/home/hybridsynchaadl/Desktop/FourDronesSystem_Forma

Options

☐ Search for nested projects

☑ Copy projects into workspace

☐ Close newly imported projects upon completion

☐ Hide projects that already exist in the workspace

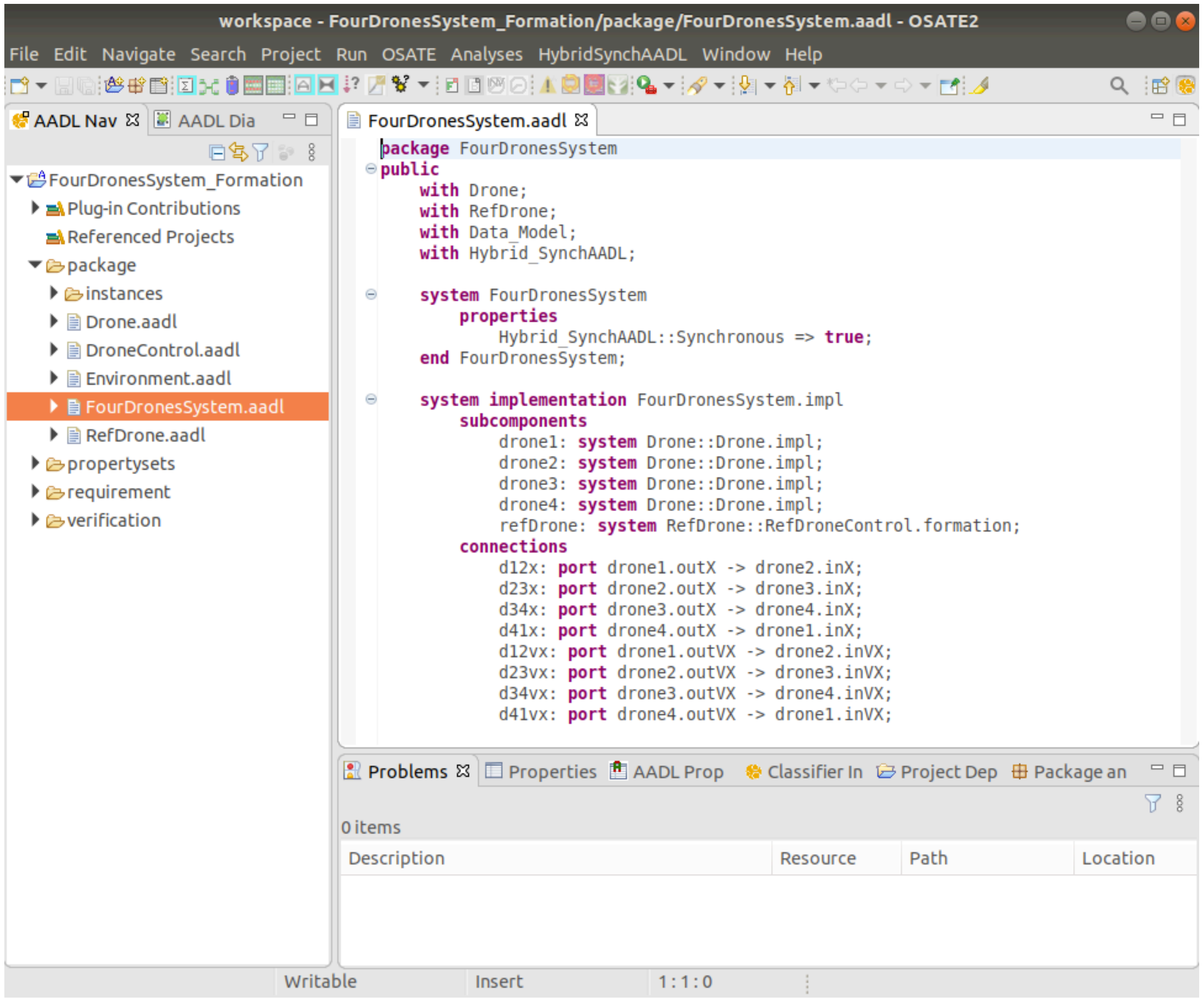
Working sets

☐ Add project to working sets

Working sets:

FourDronesSystem – Text

- `FourDroneSystem.aadl` contains the top-level system component.

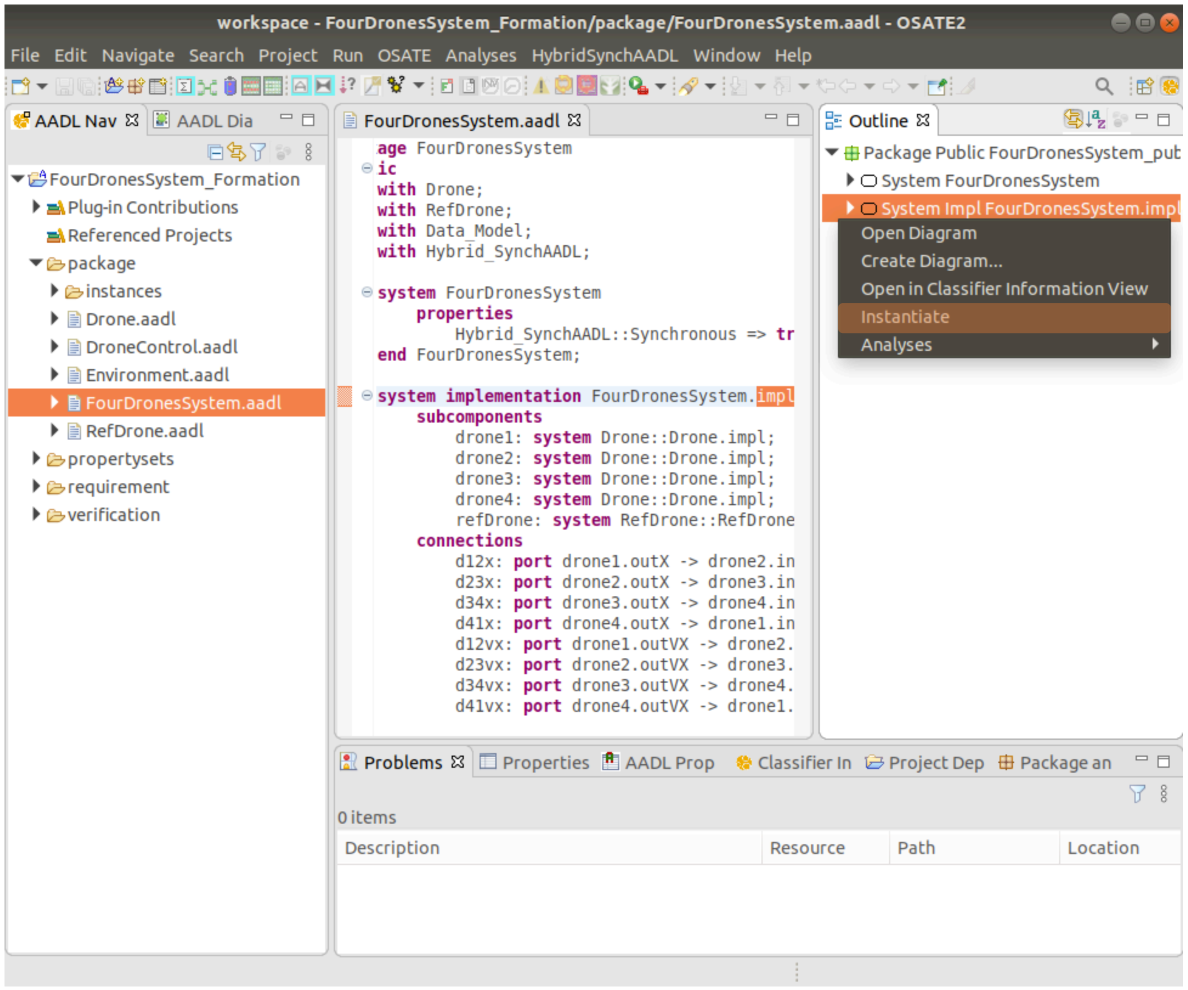


```
package FourDronesSystem
public
    with Drone;
    with RefDrone;
    with Data_Model;
    with Hybrid_SynchAADL;

    system FourDronesSystem
        properties
            Hybrid_SynchAADL::Synchronous => true;
    end FourDronesSystem;

    system implementation FourDronesSystem.impl
        subcomponents
            drone1: system Drone::Drone.impl;
            drone2: system Drone::Drone.impl;
            drone3: system Drone::Drone.impl;
            drone4: system Drone::Drone.impl;
            refDrone: system RefDrone::RefDroneControl.formation;
        connections
            d12x: port drone1.outX -> drone2.inX;
            d23x: port drone2.outX -> drone3.inX;
            d34x: port drone3.outX -> drone4.inX;
            d41x: port drone4.outX -> drone1.inX;
            d12vx: port drone1.outVX -> drone2.inVX;
            d23vx: port drone2.outVX -> drone3.inVX;
            d34vx: port drone3.outVX -> drone4.inVX;
            d41vx: port drone4.outVX -> drone1.inVX;
```

Instance Model

- Open the Outline view by clicking Menu → Window → Show view → Outline.

- Create an instance model from a system implementation as follows:
  - Right click on `System Impl FourDronesSystem.impl` and choose `Instantiate`.



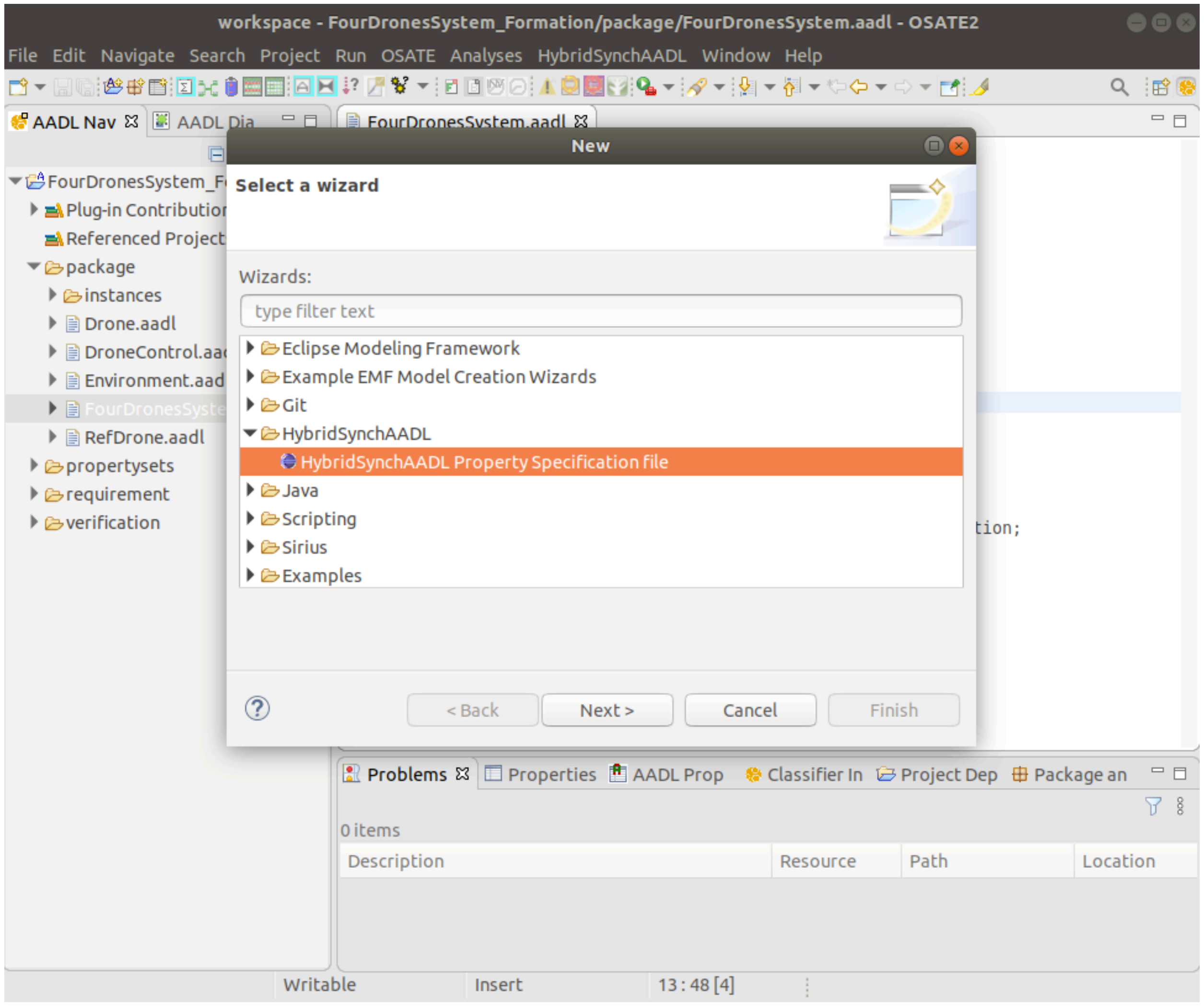
```
age FourDronesSystem
ic
with Drone;
with RefDrone;
with Data_Model;
with Hybrid_SynchAADL;

system FourDronesSystem
    properties
        Hybrid_SynchAADL::Synchronous => tr
end FourDronesSystem;

system implementation FourDronesSystem.impl
    subcomponents
        drone1: system Drone::Drone.impl;
        drone2: system Drone::Drone.impl;
        drone3: system Drone::Drone.impl;
        drone4: system Drone::Drone.impl;
        refDrone: system RefDrone::RefDrone
    connections
        d12x: port drone1.outX -> drone2.in
        d23x: port drone2.outX -> drone3.in
        d34x: port drone3.outX -> drone4.in
        d41x: port drone4.outX -> drone1.in
        d12vx: port drone1.outVX -> drone2.
        d23vx: port drone2.outVX -> drone3.
        d34vx: port drone3.outVX -> drone4.
        d41vx: port drone4.outVX -> drone1.
```

Outline

1. Basic OSATE
2. Creating Property Specification Files (PSPC)
3. HybridSynchAADL Constraints Checker
4. Maude Code Generation
5. Formal Analysis

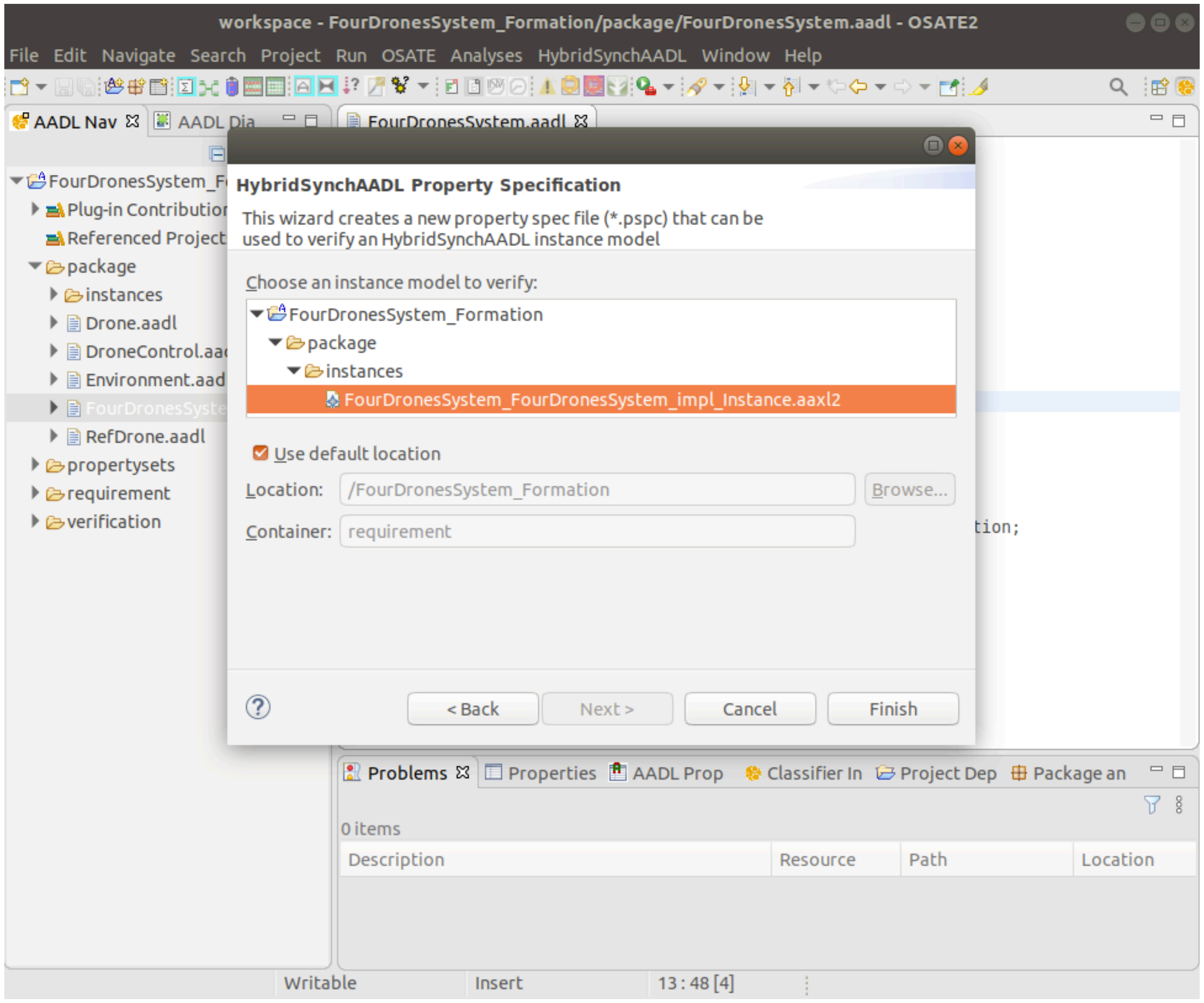
Creating PSPC Files

- To create a PSPC file, choose
  - Menu → File → New → Other → HybridSynchAADL → HybridSynchAADL Property Specification file.



Creating PSPC Files

- Any valid AADL instance model can be chosen in the wizard.
- Choose the instance model we have created in the previous slides.


workspace - FourDronesSystem_Formation/package/FourDronesSystem.aadl - OSATE2

HybridSynchAADL Property Specification

This wizard creates a new property spec file (*.pspc) that can be used to verify an HybridSynchAADL instance model

Choose an instance model to verify:

- FourDronesSystem_Formation
  - package
    - instances
      - FourDronesSystem_FourDronesSystem_impl_Instance.aaxl2

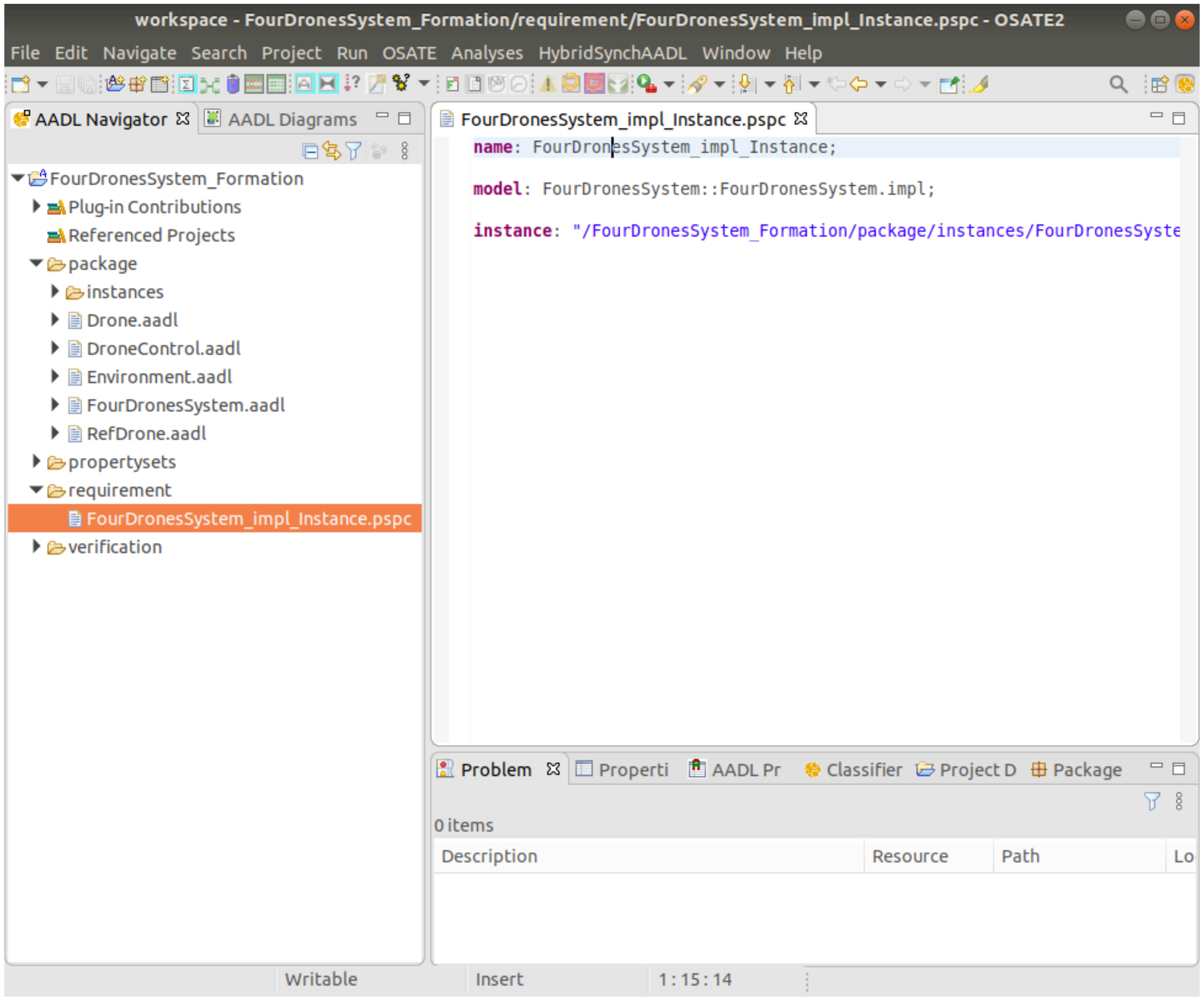
Use default location: ✓

Location: /FourDronesSystem_Formation

Container: requirement

Creating PSPC Files

- This screen shows the generated (empty) PSPC file.
- There are sample PSPC file in this project



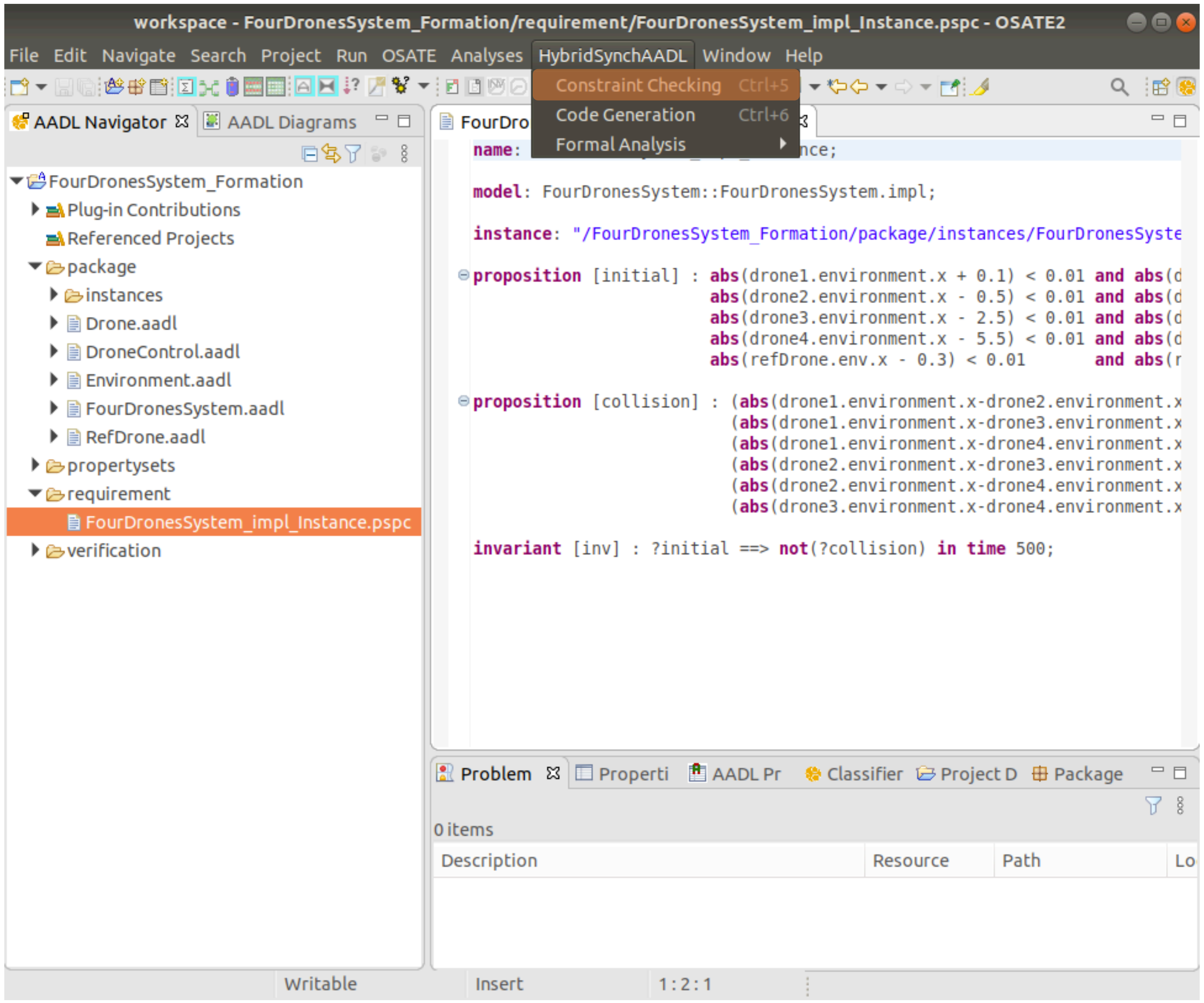
```
name: FourDronesSystem_impl_Instance;

model: FourDronesSystem::FourDronesSystem.impl;

instance: "/FourDronesSystem_Formation/package/instances/FourDronesSyste
```

Outline

1. Basic OSATE
2. Creating Property Specification Files (PSPC)
3. HybridSynchAADL Constraints Checker
4. Maude Code Generation
5. Formal Analysis

Checking HybridSynchAADL Constraints

- There are three menu items in HybridSynchAADL: `Constraints Check`, `Code Generation`, and `Formal Analysis`.
- Click `Constraints Check` to perform constraints checking.

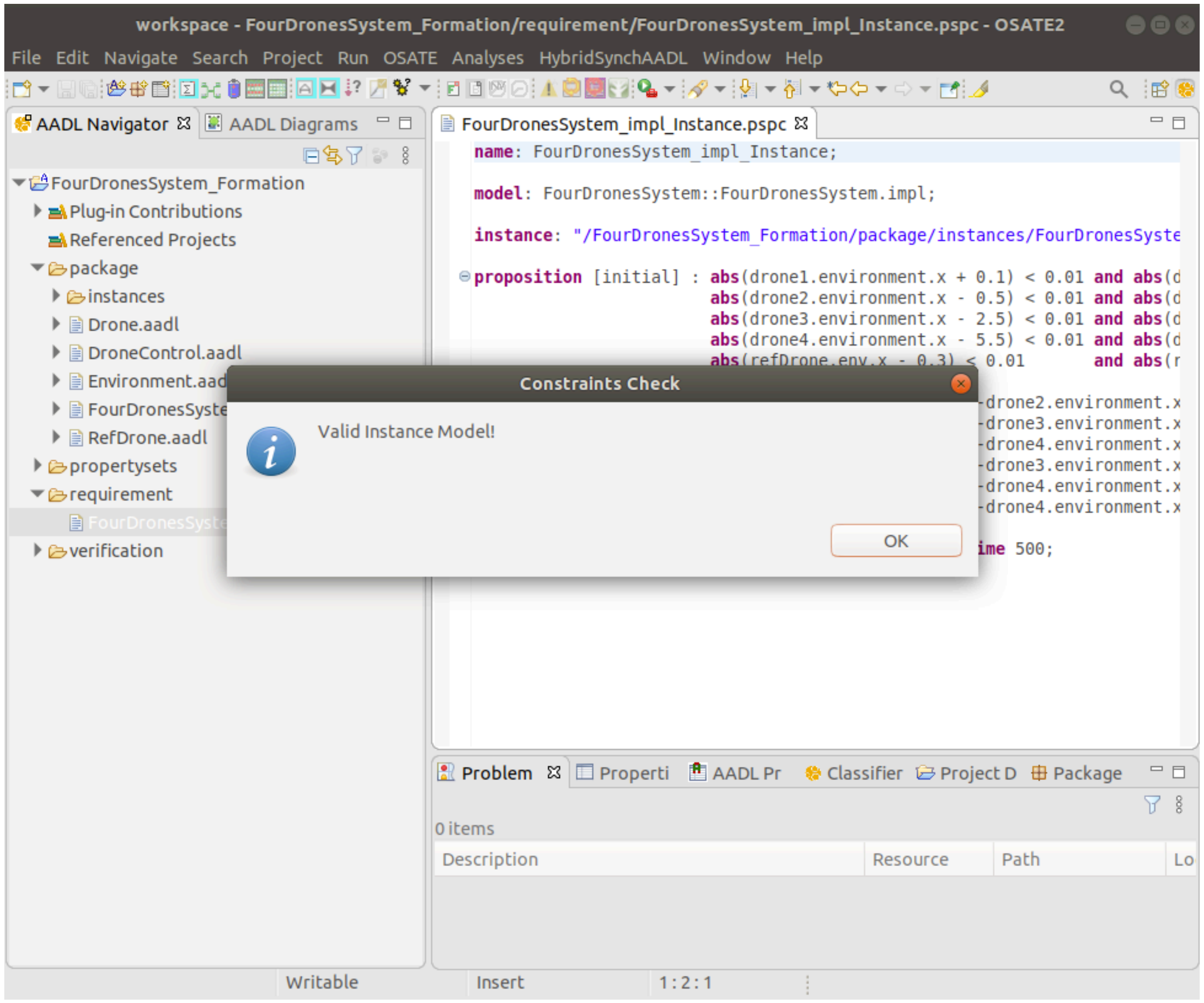


workspace - FourDronesSystem_Formation/requirement/FourDronesSystem_impl_Instance.pspc - OSATE2

```
name:                        nce;

model: FourDronesSystem::FourDronesSystem.impl;

instance: "/FourDronesSystem_Formation/package/instances/FourDronesSyste

proposition [initial] : abs(drone1.environment.x + 0.1) < 0.01 and abs(d
                        abs(drone2.environment.x - 0.5) < 0.01 and abs(d
                        abs(drone3.environment.x - 2.5) < 0.01 and abs(d
                        abs(drone4.environment.x - 5.5) < 0.01 and abs(d
                        abs(refDrone.env.x - 0.3) < 0.01       and abs(r

proposition [collision] : (abs(drone1.environment.x-drone2.environment.x
                          (abs(drone1.environment.x-drone3.environment.x
                          (abs(drone1.environment.x-drone4.environment.x
                          (abs(drone2.environment.x-drone3.environment.x
                          (abs(drone2.environment.x-drone4.environment.x
                          (abs(drone3.environment.x-drone4.environment.x

invariant [inv] : ?initial ==> not(?collision) in time 500;
```

Checking HybridSynchAADL Constraints

- When the model has no constraints error, the tool notifies that the model is valid.



```
name: FourDronesSystem_impl_Instance;

model: FourDronesSystem::FourDronesSystem.impl;

instance: "/FourDronesSystem_Formation/package/instances/FourDronesSyste

proposition [initial] : abs(drone1.environment.x + 0.1) < 0.01 and abs(d
                        abs(drone2.environment.x - 0.5) < 0.01 and abs(d
                        abs(drone3.environment.x - 2.5) < 0.01 and abs(d
                        abs(drone4.environment.x - 5.5) < 0.01 and abs(d
                        abs(refDrone.env.x - 0.3) < 0.01       and abs(r

                                                          -drone2.environment.x
                                                          -drone3.environment.x
                                                          -drone4.environment.x
                                                          -drone3.environment.x
                                                          -drone4.environment.x
                                                          -drone4.environment.x

                                                          ime 500;
```

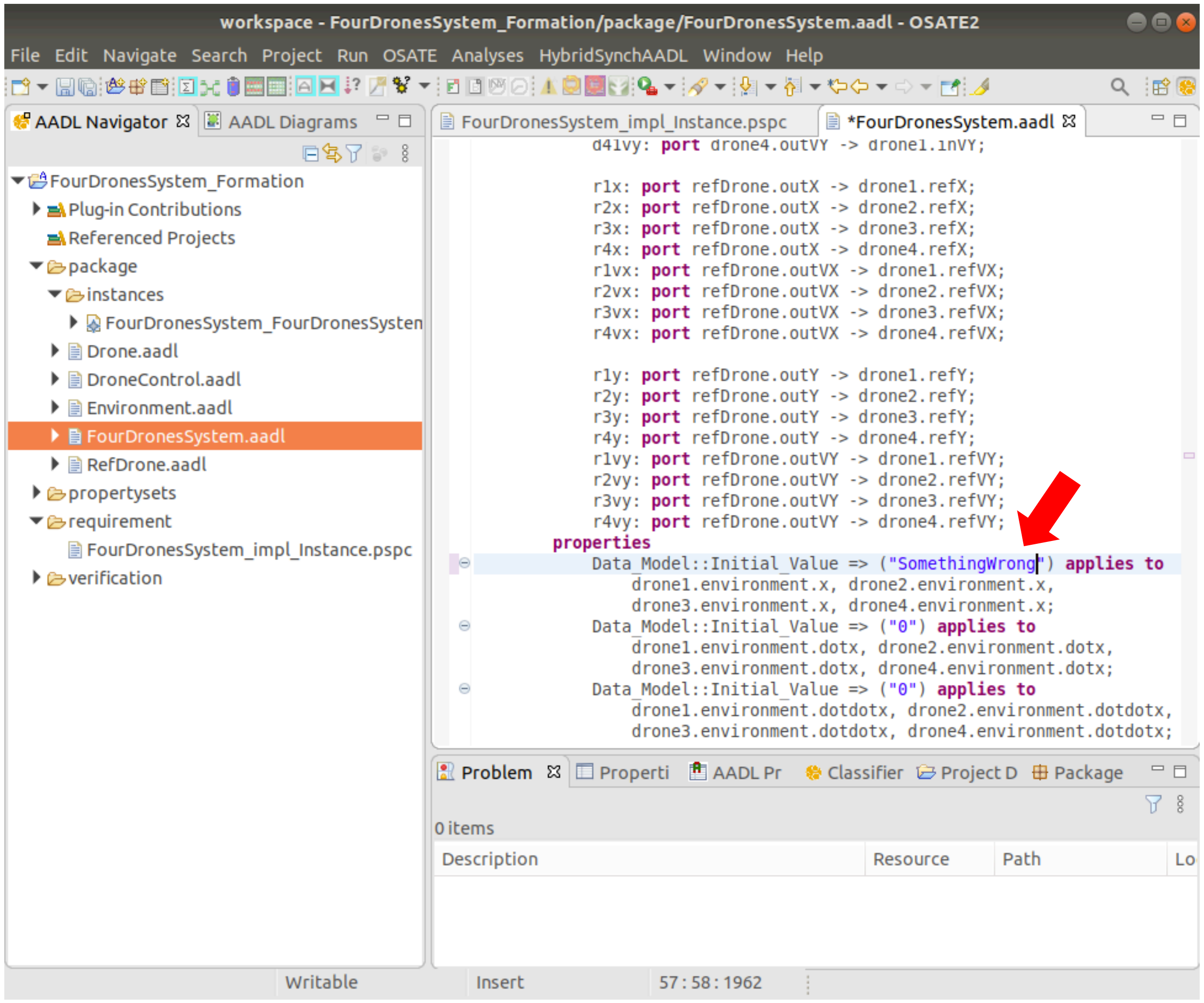
Constraints Check

Valid Instance Model!

OK

Constraints Check – Erroneous Model

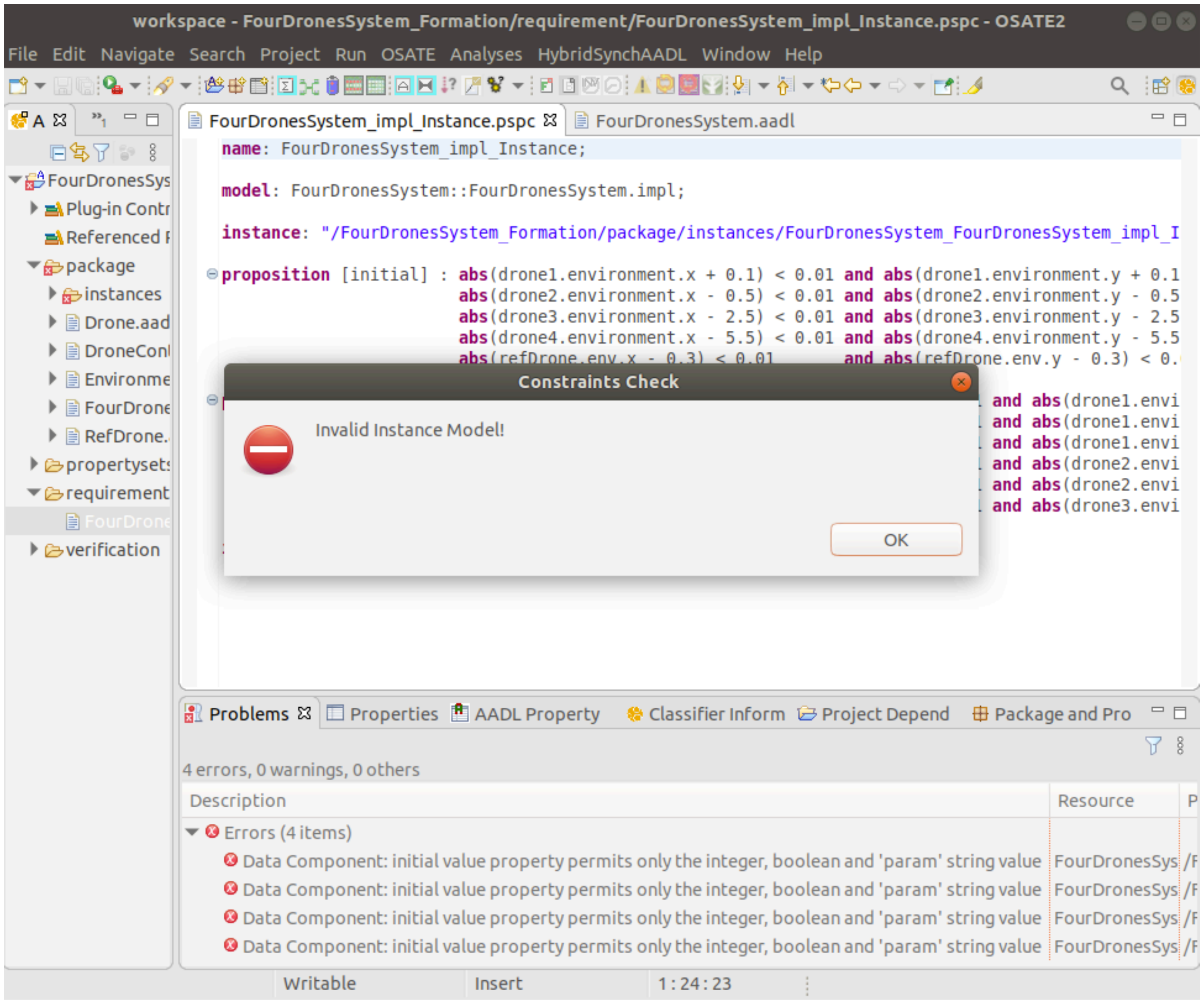
- What if some HybridSynchAADL constraints is not satisfied?
- Let us add an invalid initial value to data component and see what happened.
    - by changing the property value `param => SomethingWrong.`



```
            d41vy: port drone4.outVY -> drone1.inVY;

            r1x: port refDrone.outX -> drone1.refX;
            r2x: port refDrone.outX -> drone2.refX;
            r3x: port refDrone.outX -> drone3.refX;
            r4x: port refDrone.outX -> drone4.refX;
            r1vx: port refDrone.outVX -> drone1.refVX;
            r2vx: port refDrone.outVX -> drone2.refVX;
            r3vx: port refDrone.outVX -> drone3.refVX;
            r4vx: port refDrone.outVX -> drone4.refVX;

            r1y: port refDrone.outY -> drone1.refY;
            r2y: port refDrone.outY -> drone2.refY;
            r3y: port refDrone.outY -> drone3.refY;
            r4y: port refDrone.outY -> drone4.refY;
            r1vy: port refDrone.outVY -> drone1.refVY;
            r2vy: port refDrone.outVY -> drone2.refVY;
            r3vy: port refDrone.outVY -> drone3.refVY;
            r4vy: port refDrone.outVY -> drone4.refVY;
        properties
            Data_Model::Initial_Value => ("SomethingWrong") applies to
                drone1.environment.x, drone2.environment.x,
                drone3.environment.x, drone4.environment.x;
            Data_Model::Initial_Value => ("0") applies to
                drone1.environment.dotx, drone2.environment.dotx,
                drone3.environment.dotx, drone4.environment.dotx;
            Data_Model::Initial_Value => ("0") applies to
                drone1.environment.dotdotx, drone2.environment.dotdotx,
                drone3.environment.dotdotx, drone4.environment.dotdotx;
```

Constraints Check – Erroneous Model

- After re-instantiating the model, click `Constraints Check` to perform constraints checking.
- Click `Initial Mode`
- Our tool then shows an error message in the Problems view.


workspace - FourDronesSystem_Formation/requirement/FourDronesSystem_impl_Instance.pspc - OSATE2

```
name: FourDronesSystem_impl_Instance;

model: FourDronesSystem::FourDronesSystem.impl;

instance: "/FourDronesSystem_Formation/package/instances/FourDronesSystem_FourDronesSystem_impl_I

proposition [initial] : abs(drone1.environment.x + 0.1) < 0.01 and abs(drone1.environment.y + 0.1
                        abs(drone2.environment.x - 0.5) < 0.01 and abs(drone2.environment.y - 0.5
                        abs(drone3.environment.x - 2.5) < 0.01 and abs(drone3.environment.y - 2.5
                        abs(drone4.environment.x - 5.5) < 0.01 and abs(drone4.environment.y - 5.5
                        abs(refDrone.env.x - 0.3) < 0.01       and abs(refDrone.env.y - 0.3) < 0.
                                                               and abs(drone1.envi
                                                               and abs(drone1.envi
                                                               and abs(drone1.envi
                                                               and abs(drone2.envi
                                                               and abs(drone2.envi
                                                               and abs(drone3.envi
```

Constraints Check

Invalid Instance Model!

4 errors, 0 warnings, 0 others

Description	Resource	P
Errors (4 items)		
Data Component: initial value property permits only the integer, boolean and 'param' string value	FourDronesSys	/F
Data Component: initial value property permits only the integer, boolean and 'param' string value	FourDronesSys	/F
Data Component: initial value property permits only the integer, boolean and 'param' string value	FourDronesSys	/F
Data Component: initial value property permits only the integer, boolean and 'param' string value	FourDronesSys	/F

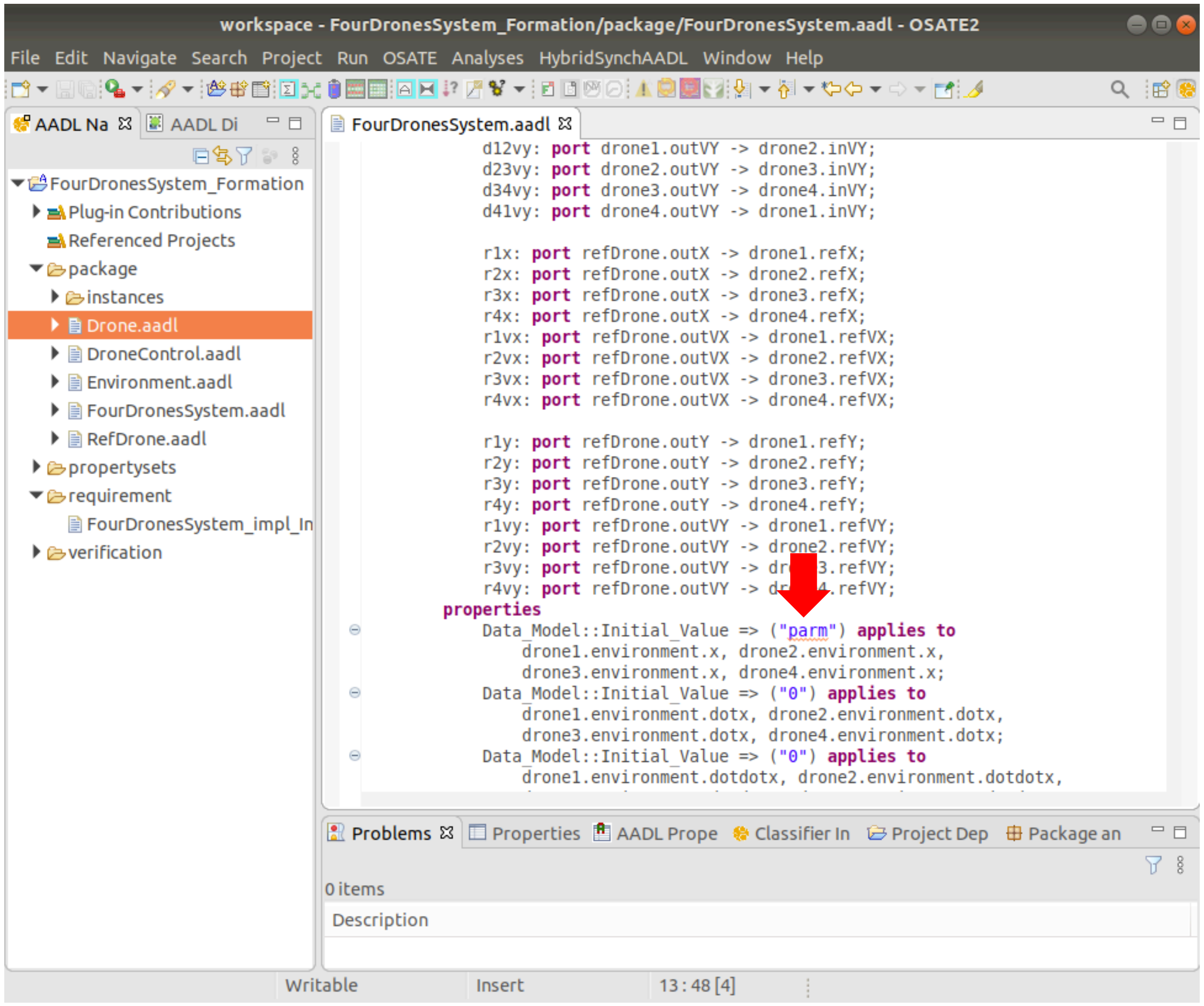
Writable   Insert   1 : 24 : 23

Outline

1. Basic OSATE
2. Creating Property Specification Files (PSPC)
3. HybridSynchAADL Constraints Checker
4. Maude Code Generation
5. Formal Analysis

The FourDronesSystem Example

- Let us go back to the correct model.
- Don't forget to instantiate the model again.

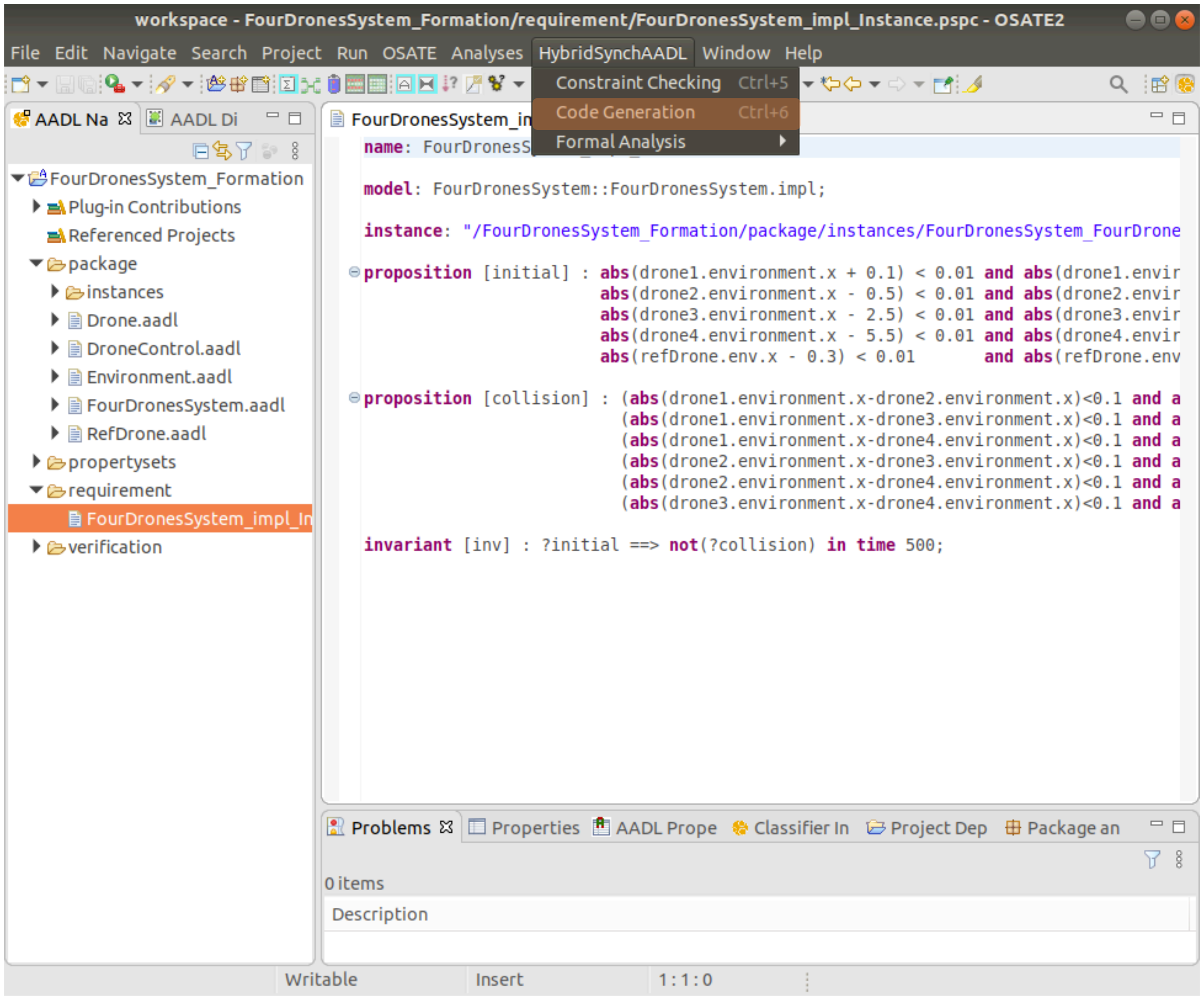


workspace - FourDronesSystem_Formation/package/FourDronesSystem.aadl - OSATE2

```
        d12vy: port drone1.outVY -> drone2.inVY;
        d23vy: port drone2.outVY -> drone3.inVY;
        d34vy: port drone3.outVY -> drone4.inVY;
        d41vy: port drone4.outVY -> drone1.inVY;

        r1x: port refDrone.outX -> drone1.refX;
        r2x: port refDrone.outX -> drone2.refX;
        r3x: port refDrone.outX -> drone3.refX;
        r4x: port refDrone.outX -> drone4.refX;
        r1vx: port refDrone.outVX -> drone1.refVX;
        r2vx: port refDrone.outVX -> drone2.refVX;
        r3vx: port refDrone.outVX -> drone3.refVX;
        r4vx: port refDrone.outVX -> drone4.refVX;

        r1y: port refDrone.outY -> drone1.refY;
        r2y: port refDrone.outY -> drone2.refY;
        r3y: port refDrone.outY -> drone3.refY;
        r4y: port refDrone.outY -> drone4.refY;
        r1vy: port refDrone.outVY -> drone1.refVY;
        r2vy: port refDrone.outVY -> drone2.refVY;
        r3vy: port refDrone.outVY -> drone3.refVY;
        r4vy: port refDrone.outVY -> drone4.refVY;
    properties
        Data_Model::Initial_Value => ("parm") applies to
            drone1.environment.x, drone2.environment.x,
            drone3.environment.x, drone4.environment.x;
        Data_Model::Initial_Value => ("0") applies to
            drone1.environment.dotx, drone2.environment.dotx,
            drone3.environment.dotx, drone4.environment.dotx;
        Data_Model::Initial_Value => ("0") applies to
            drone1.environment.dotdotx, drone2.environment.dotdotx,
```

Maude Code Generation

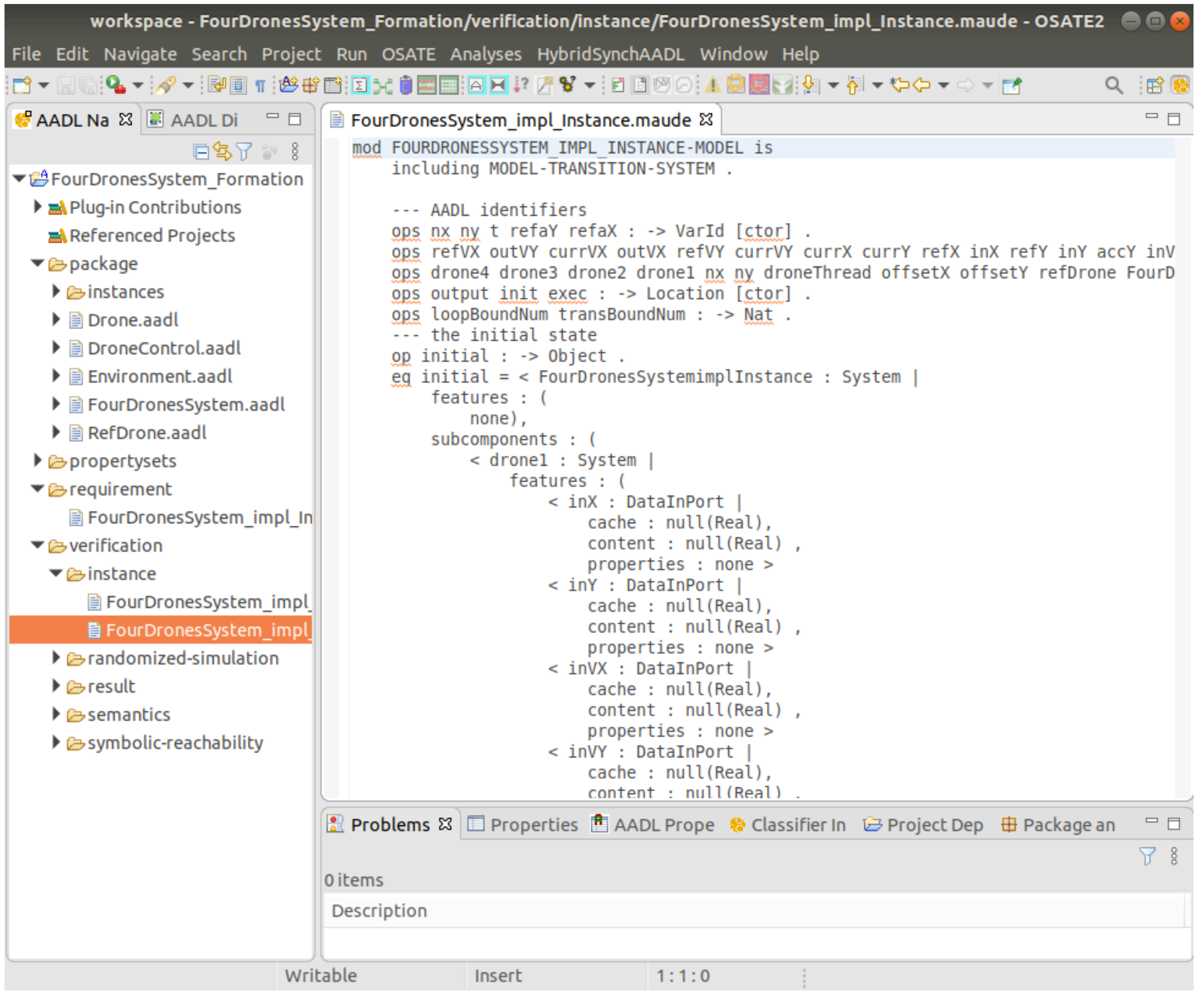
- Click `Code Generation` to automatically generate the rewriting-modulo-SMT model from the HybridSynchAADL model.



```
name: FourDronesS

model: FourDronesSystem::FourDronesSystem.impl;

instance: "/FourDronesSystem_Formation/package/instances/FourDronesSystem_FourDrone

proposition [initial] : abs(drone1.environment.x + 0.1) < 0.01 and abs(drone1.envir
                        abs(drone2.environment.x - 0.5) < 0.01 and abs(drone2.envir
                        abs(drone3.environment.x - 2.5) < 0.01 and abs(drone3.envir
                        abs(drone4.environment.x - 5.5) < 0.01 and abs(drone4.envir
                        abs(refDrone.env.x - 0.3) < 0.01       and abs(refDrone.env

proposition [collision] : (abs(drone1.environment.x-drone2.environment.x)<0.1 and a
                          (abs(drone1.environment.x-drone3.environment.x)<0.1 and a
                          (abs(drone1.environment.x-drone4.environment.x)<0.1 and a
                          (abs(drone2.environment.x-drone3.environment.x)<0.1 and a
                          (abs(drone2.environment.x-drone4.environment.x)<0.1 and a
                          (abs(drone3.environment.x-drone4.environment.x)<0.1 and a

invariant [inv] : ?initial ==> not(?collision) in time 500;
```

Maude Code Generation

- The generated Maude files, including Maude files for properties, are in the `verification/instance` directory.



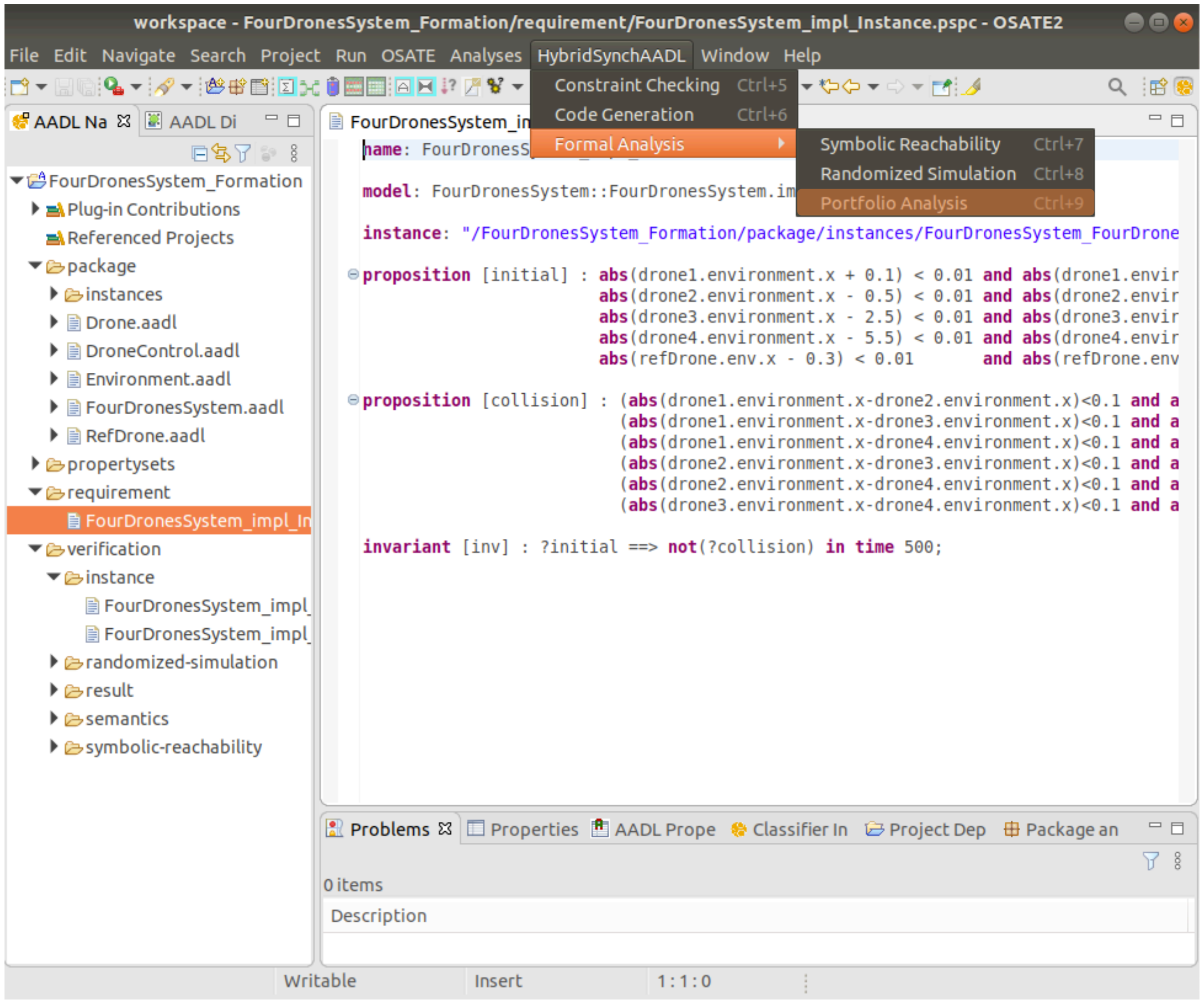
```
mod FOURDRONESSYSTEM_IMPL_INSTANCE-MODEL is
  including MODEL-TRANSITION-SYSTEM .

  --- AADL identifiers
  ops nx ny t refaY refaX : -> VarId [ctor] .
  ops refVX outVY currVX outVX refVY currVY currX currY refX inX refY inY accY inV
  ops drone4 drone3 drone2 drone1 nx ny droneThread offsetX offsetY refDrone FourD
  ops output init exec : -> Location [ctor] .
  ops loopBoundNum transBoundNum : -> Nat .
  --- the initial state
  op initial : -> Object .
  eq initial = < FourDronesSystemimplInstance : System |
      features : (
          none),
      subcomponents : (
          < drone1 : System |
              features : (
                  < inX : DataInPort |
                      cache : null(Real),
                      content : null(Real) ,
                      properties : none >
                  < inY : DataInPort |
                      cache : null(Real),
                      content : null(Real) ,
                      properties : none >
                  < inVX : DataInPort |
                      cache : null(Real),
                      content : null(Real) ,
                      properties : none >
                  < inVY : DataInPort |
                      cache : null(Real),
                      content : null(Real) .
```

Outline

1. Basic OSATE
2. Creating Property Specification Files (PSPC)
3. HybridSynchAADL Constraints Checker
4. Maude Code Generation
5. Formal Analysis

Portfolio Analysis

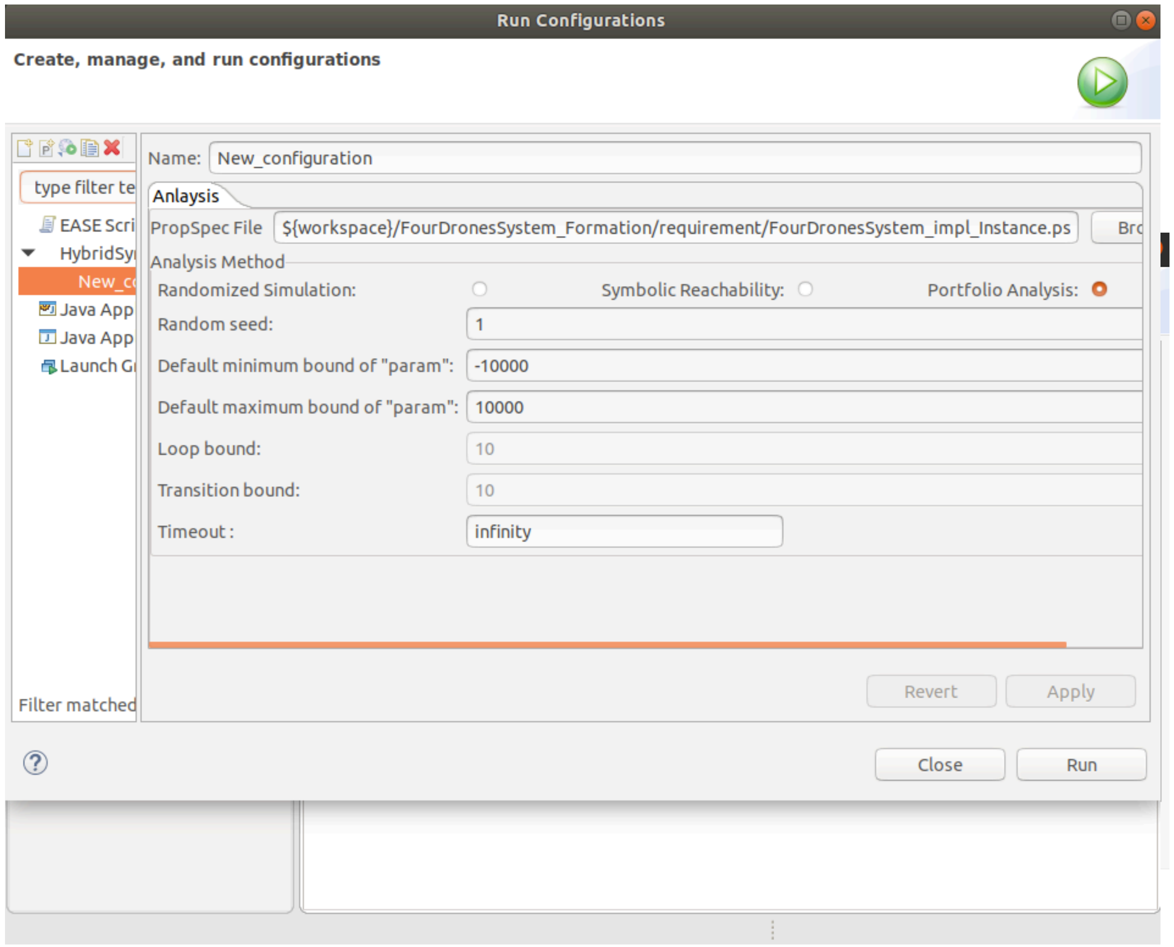
- Click `Portfolio Analysis` to perform symbolic reachability and randomized simulation simultaneously using rewriting-modulo-SMT.



```
name: FourDronesS

model: FourDronesSystem::FourDronesSystem.im

instance: "/FourDronesSystem_Formation/package/instances/FourDronesSystem_FourDrone

proposition [initial] : abs(drone1.environment.x + 0.1) < 0.01 and abs(drone1.envir
                        abs(drone2.environment.x - 0.5) < 0.01 and abs(drone2.envir
                        abs(drone3.environment.x - 2.5) < 0.01 and abs(drone3.envir
                        abs(drone4.environment.x - 5.5) < 0.01 and abs(drone4.envir
                        abs(refDrone.env.x - 0.3) < 0.01       and abs(refDrone.env

proposition [collision] : (abs(drone1.environment.x-drone2.environment.x)<0.1 and a
                          (abs(drone1.environment.x-drone3.environment.x)<0.1 and a
                          (abs(drone1.environment.x-drone4.environment.x)<0.1 and a
                          (abs(drone2.environment.x-drone3.environment.x)<0.1 and a
                          (abs(drone2.environment.x-drone4.environment.x)<0.1 and a
                          (abs(drone3.environment.x-drone4.environment.x)<0.1 and a

invariant [inv] : ?initial ==> not(?collision) in time 500;
```

Portfolio Analysis

- Create a new configuration file
- Set PSPC file "FourDronesSystem_impl_Instance1.pspc" path
- Click `Portfolio Analysis` radio button
- Set positive integer value in `Timeout`
  - `infinity` can be set for infinite time.



Name: New_configuration

Anlaysis

PropSpec File: ${workspace}/FourDronesSystem_Formation/requirement/FourDronesSystem_impl_Instance.ps

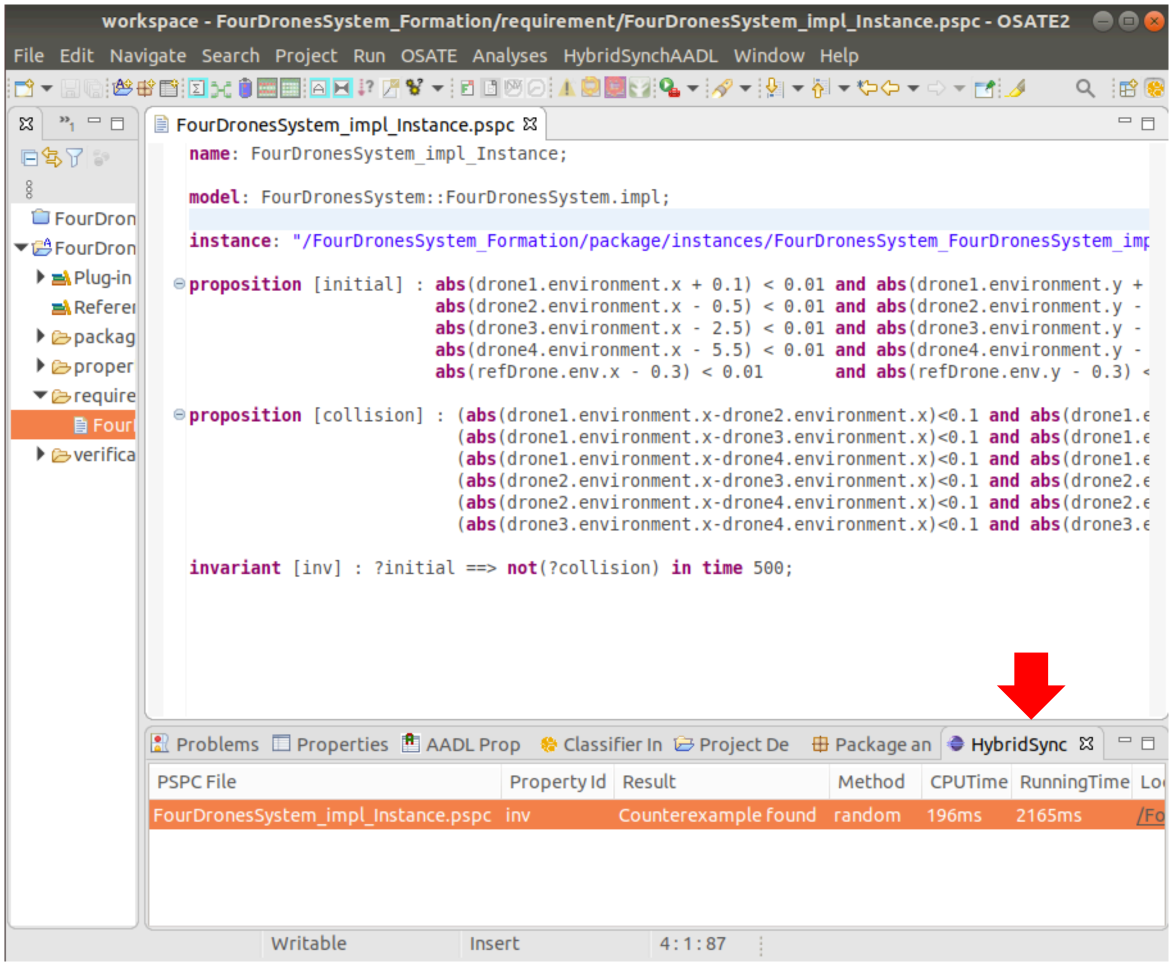
Analysis Method

Randomized Simulation: ○ Symbolic Reachability: ○ Portfolio Analysis: [✓ ●]

Random seed:	1
Default minimum bound of "param":	-10000
Default maximum bound of "param":	10000
Loop bound:	10
Transition bound:	10
Timeout :	infinity

Analysis Results

- The HybridSynchAADL Result view shows the analysis results.

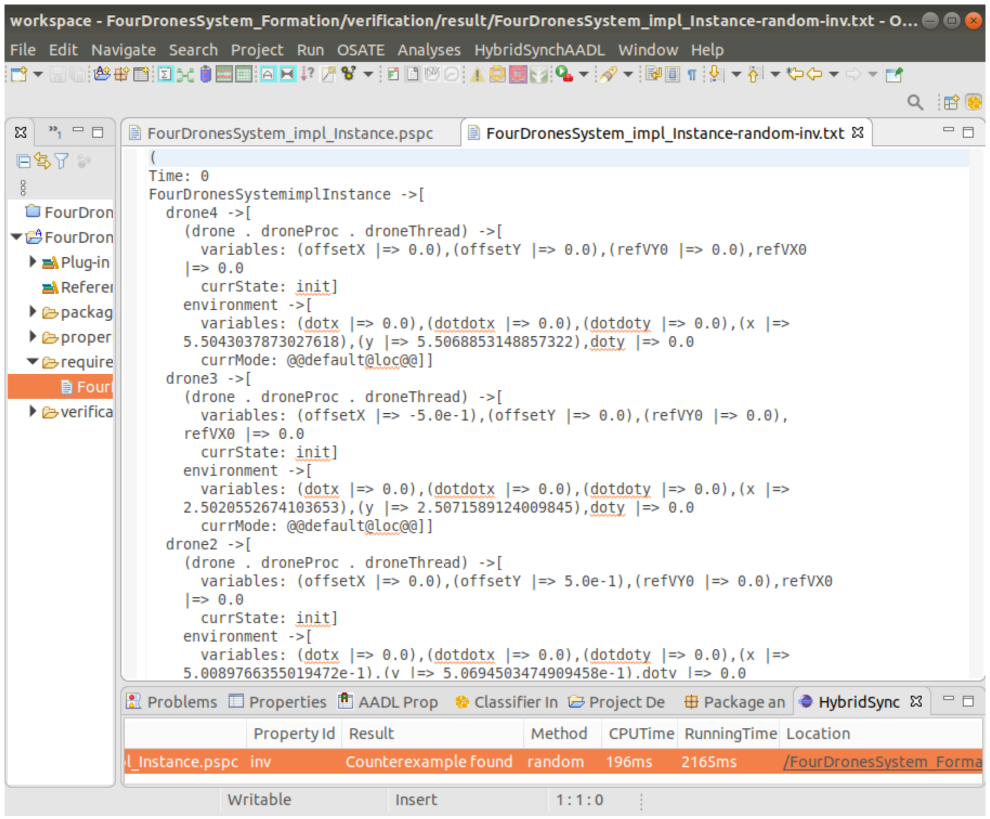

workspace - FourDronesSystem_Formation/requirement/FourDronesSystem_impl_Instance.pspc - OSATE2
File Edit Navigate Search Project Run OSATE Analyses HybridSynchAADL Window Help
FourDron
FourDron
Plug-in
Referer
packag
proper
require
Four
verifica
FourDronesSystem_impl_Instance.pspc
Problems Properties AADL Prop Classifier In Project De Package an HybridSync
Writable Insert 4:1:87

```
name: FourDronesSystem_impl_Instance;

model: FourDronesSystem::FourDronesSystem.impl;

instance: "/FourDronesSystem_Formation/package/instances/FourDronesSystem_FourDronesSystem_imp

proposition [initial] : abs(drone1.environment.x + 0.1) < 0.01 and abs(drone1.environment.y +
                        abs(drone2.environment.x - 0.5) < 0.01 and abs(drone2.environment.y -
                        abs(drone3.environment.x - 2.5) < 0.01 and abs(drone3.environment.y -
                        abs(drone4.environment.x - 5.5) < 0.01 and abs(drone4.environment.y -
                        abs(refDrone.env.x - 0.3) < 0.01       and abs(refDrone.env.y - 0.3) <

proposition [collision] : (abs(drone1.environment.x-drone2.environment.x)<0.1 and abs(drone1.e
                          (abs(drone1.environment.x-drone3.environment.x)<0.1 and abs(drone1.e
                          (abs(drone1.environment.x-drone4.environment.x)<0.1 and abs(drone1.e
                          (abs(drone2.environment.x-drone3.environment.x)<0.1 and abs(drone2.e
                          (abs(drone2.environment.x-drone4.environment.x)<0.1 and abs(drone2.e
                          (abs(drone3.environment.x-drone4.environment.x)<0.1 and abs(drone3.e

invariant [inv] : ?initial ==> not(?collision) in time 500;
```

PSPC File	Property Id	Result	Method	CPUTime	RunningTime	Lo
FourDronesSystem_impl_Instance.pspc	inv	Counterexample found	random	196ms	2165ms	/Fo

Counterexample

- Each file in Location in the result view contains a counterexample of an invariant property if it exists.



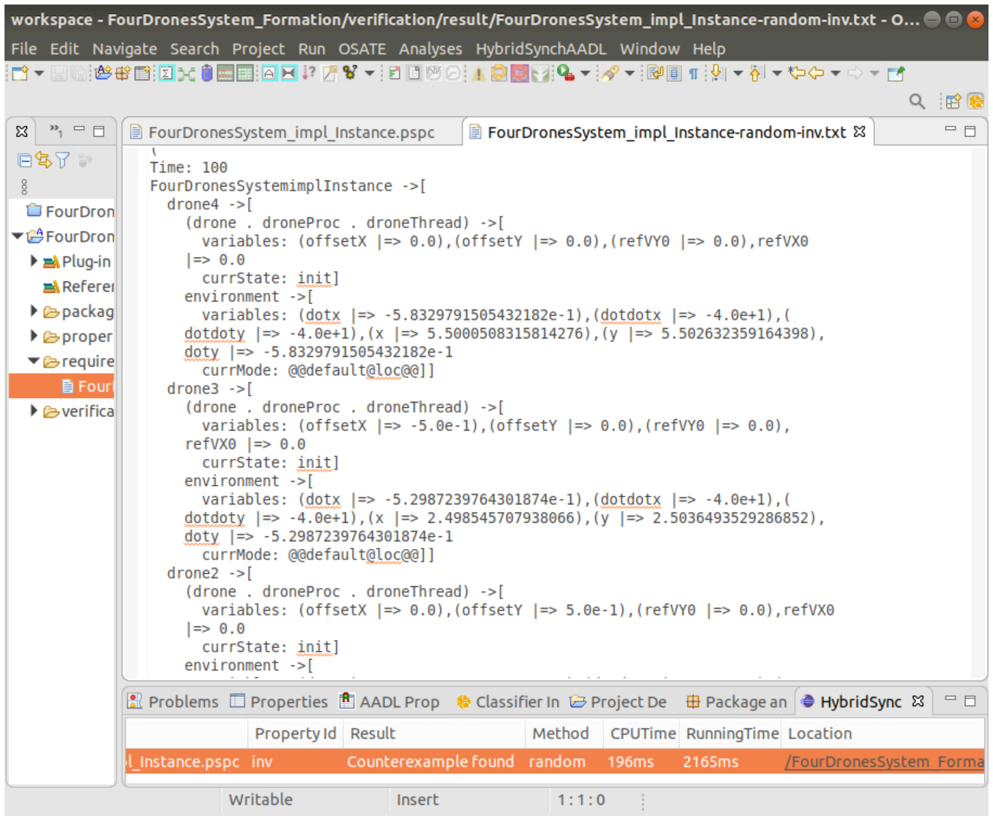
workspace - FourDronesSystem_Formation/verification/result/FourDronesSystem_impl_Instance-random-inv.txt - O...

FourDronesSystem_impl_Instance.pspc | FourDronesSystem_impl_Instance-random-inv.txt

```
(
Time: 0
FourDronesSystemimplInstance ->[
  drone4 ->[
    (drone . droneProc . droneThread) ->[
      variables: (offsetX |=> 0.0),(offsetY |=> 0.0),(refVY0 |=> 0.0),refVX0
    |=> 0.0
      currState: init]
    environment ->[
      variables: (dotx |=> 0.0),(dotdotx |=> 0.0),(dotdoty |=> 0.0),(x |=>
    5.5043037873027618),(y |=> 5.5068853148857322),doty |=> 0.0
      currMode: @@default@loc@@]]
  drone3 ->[
    (drone . droneProc . droneThread) ->[
      variables: (offsetX |=> -5.0e-1),(offsetY |=> 0.0),(refVY0 |=> 0.0),
    refVX0 |=> 0.0
      currState: init]
    environment ->[
      variables: (dotx |=> 0.0),(dotdotx |=> 0.0),(dotdoty |=> 0.0),(x |=>
    2.5020552674103653),(y |=> 2.5071589124009845),doty |=> 0.0
      currMode: @@default@loc@@]]
  drone2 ->[
    (drone . droneProc . droneThread) ->[
      variables: (offsetX |=> 0.0),(offsetY |=> 5.0e-1),(refVY0 |=> 0.0),refVX0
    |=> 0.0
      currState: init]
    environment ->[
      variables: (dotx |=> 0.0),(dotdotx |=> 0.0),(dotdoty |=> 0.0),(x |=>
    5.0089766355019472e-1),(y |=> 5.0694503474909458e-1),doty |=> 0.0
```

	Property Id	Result	Method	CPUTime	RunningTime	Location
l_Instance.pspc	inv	Counterexample found	random	196ms	2165ms	/FourDronesSystem_Forma

Writable | Insert | 1 : 1 : 0



workspace - FourDronesSystem_Formation/verification/result/FourDronesSystem_impl_Instance-random-inv.txt - O...

FourDronesSystem_impl_Instance.pspc | FourDronesSystem_impl_Instance-random-inv.txt

```
Time: 100
FourDronesSystemimplInstance ->[
  drone4 ->[
    (drone . droneProc . droneThread) ->[
      variables: (offsetX |=> 0.0),(offsetY |=> 0.0),(refVY0 |=> 0.0),refVX0
    |=> 0.0
      currState: init]
    environment ->[
      variables: (dotx |=> -5.8329791505432182e-1),(dotdotx |=> -4.0e+1),(
    dotdoty |=> -4.0e+1),(x |=> 5.5000508315814276),(y |=> 5.502632359164398),
    doty |=> -5.8329791505432182e-1
      currMode: @@default@loc@@]]
  drone3 ->[
    (drone . droneProc . droneThread) ->[
      variables: (offsetX |=> -5.0e-1),(offsetY |=> 0.0),(refVY0 |=> 0.0),
    refVX0 |=> 0.0
      currState: init]
    environment ->[
      variables: (dotx |=> -5.2987239764301874e-1),(dotdotx |=> -4.0e+1),(
    dotdoty |=> -4.0e+1),(x |=> 2.498545707938066),(y |=> 2.5036493529286852),
    doty |=> -5.2987239764301874e-1
      currMode: @@default@loc@@]]
  drone2 ->[
    (drone . droneProc . droneThread) ->[
      variables: (offsetX |=> 0.0),(offsetY |=> 5.0e-1),(refVY0 |=> 0.0),refVX0
    |=> 0.0
      currState: init]
    environment ->[
```

	Property Id	Result	Method	CPUTime	RunningTime	Location
l_Instance.pspc	inv	Counterexample found	random	196ms	2165ms	/FourDronesSystem_Forma

Writable | Insert | 1 : 1 : 0

Thank you!