HybridSynchAADL

Tutorial

Outline

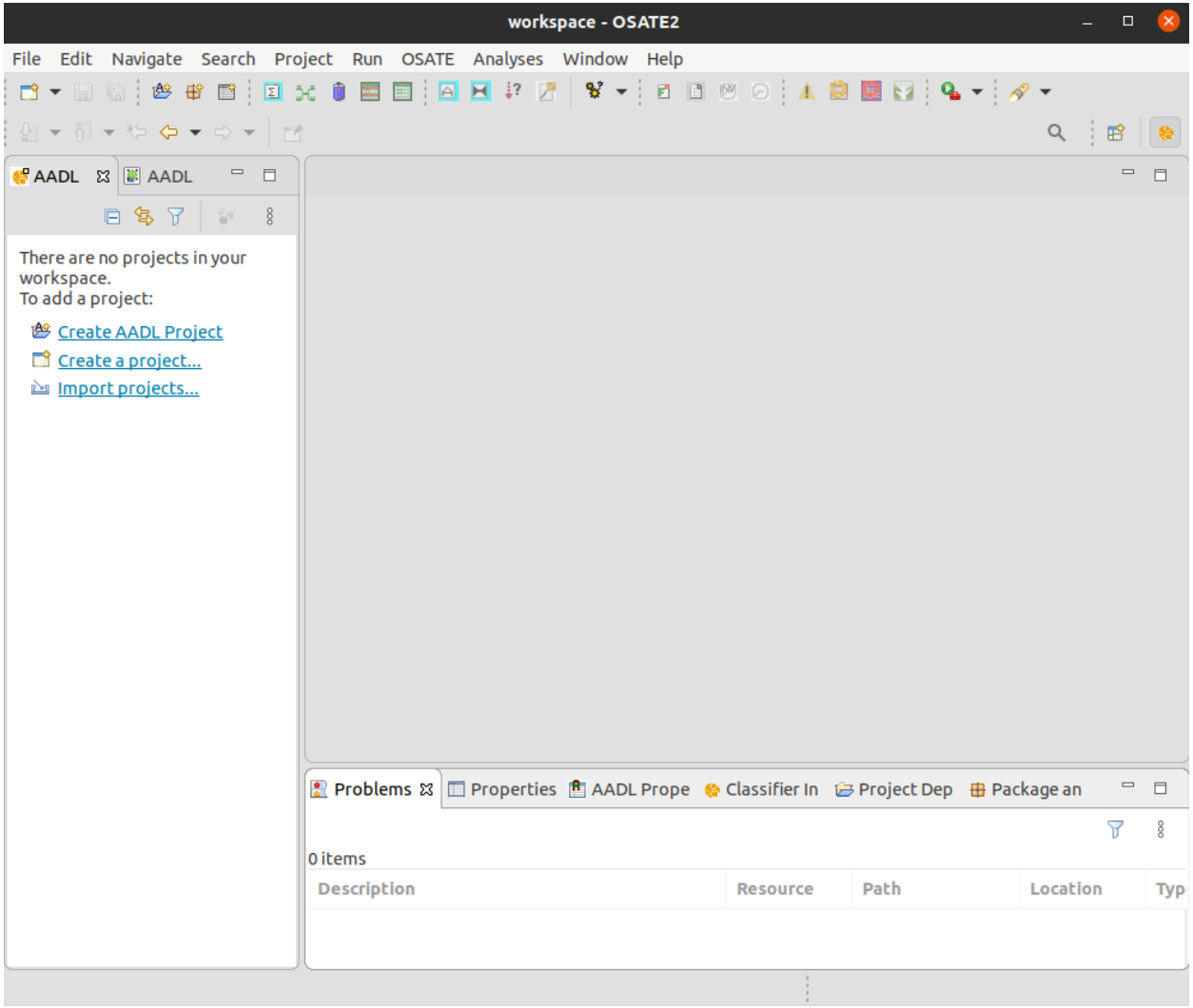
1. Basic OSATE
2. Creating Property Specification Files (PSPC)
3. HybridSynchAADL Constraints Checker
4. Maude Code Generation
5. Formal Analysis

Outline

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

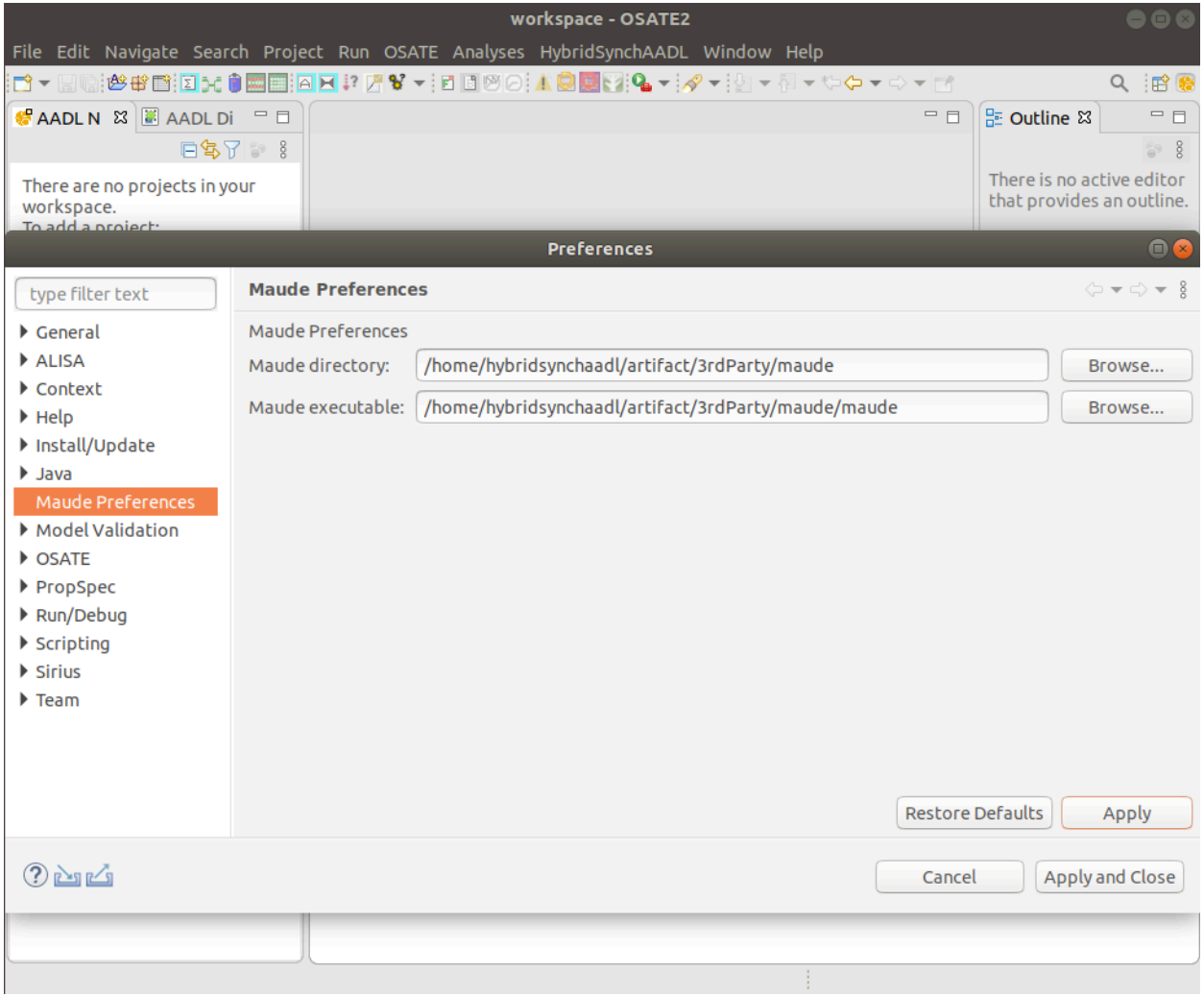
Running OSATE

- You will see this window when you execute OSATE.



Maude Preferences

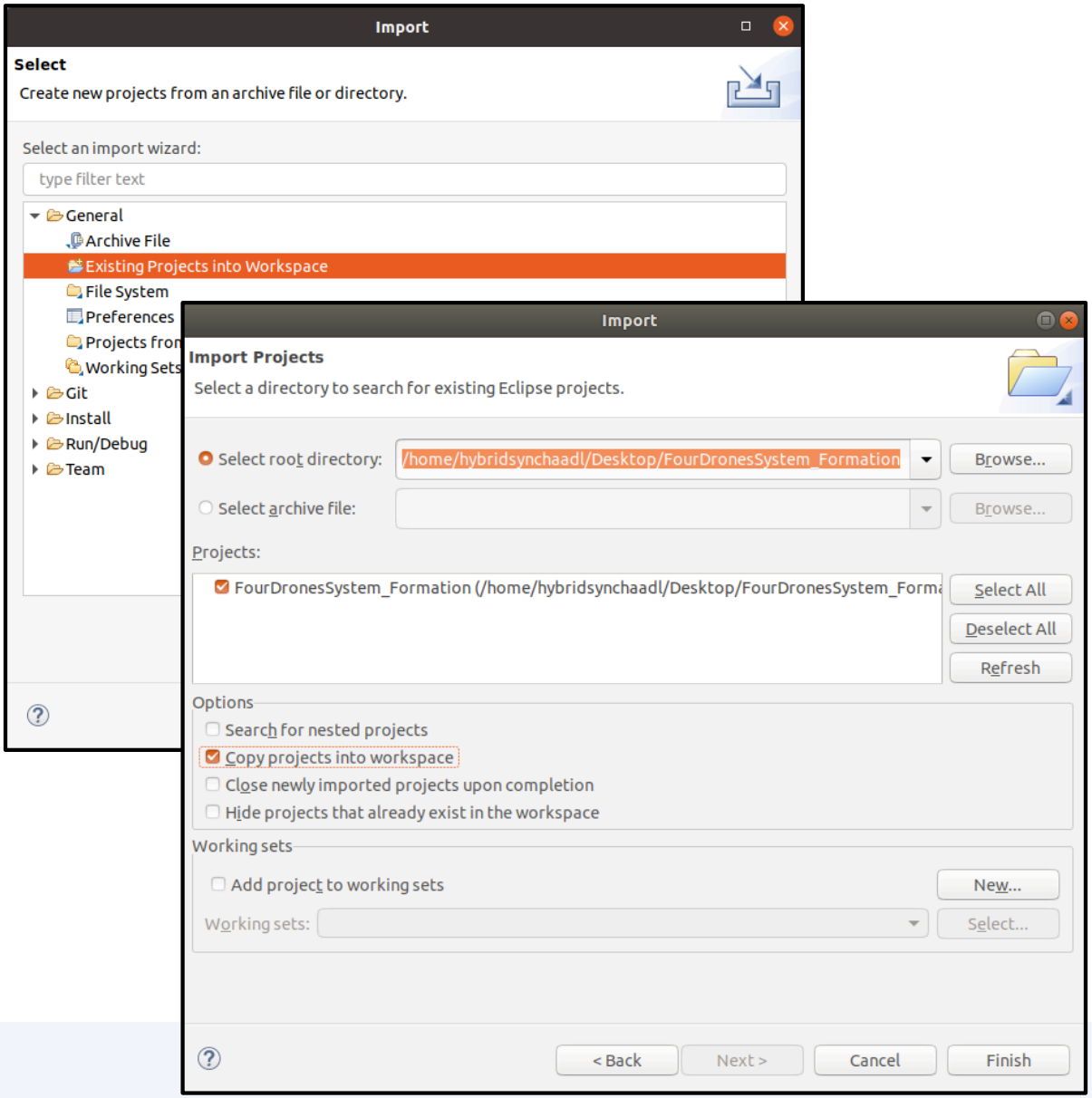
- Before importing an example project, set the proper Maude preferences.
- Open Preferences menu by clicking Menu → Window → Preferences.
- Set Maude directory and executable file location.



OSATE - Importing an Example

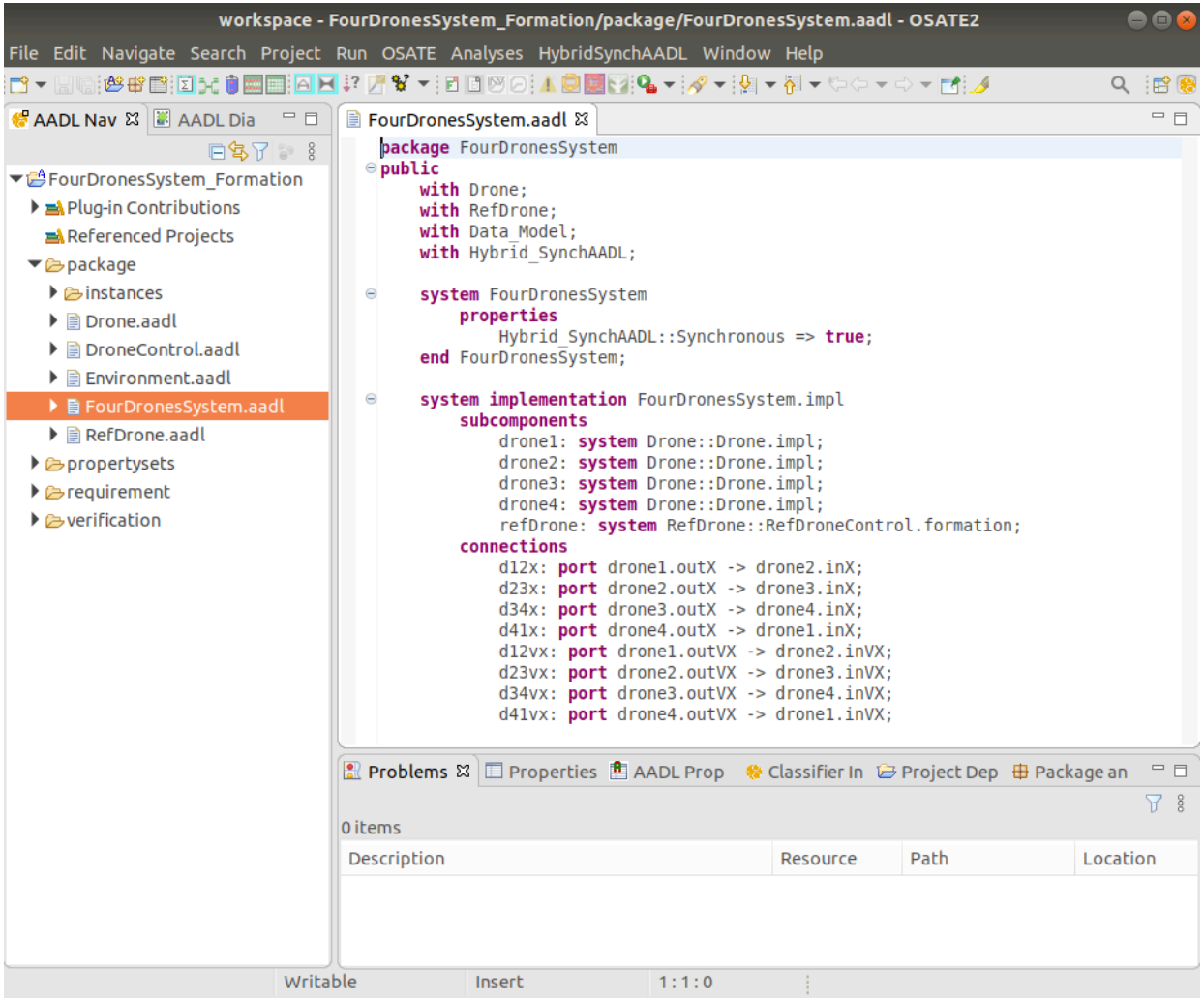
- We start with a simple example, namely, `FourDronesSystem_Formation` in the directory `models/hybridsynchaadl`.

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



FourDronesSystem – Text

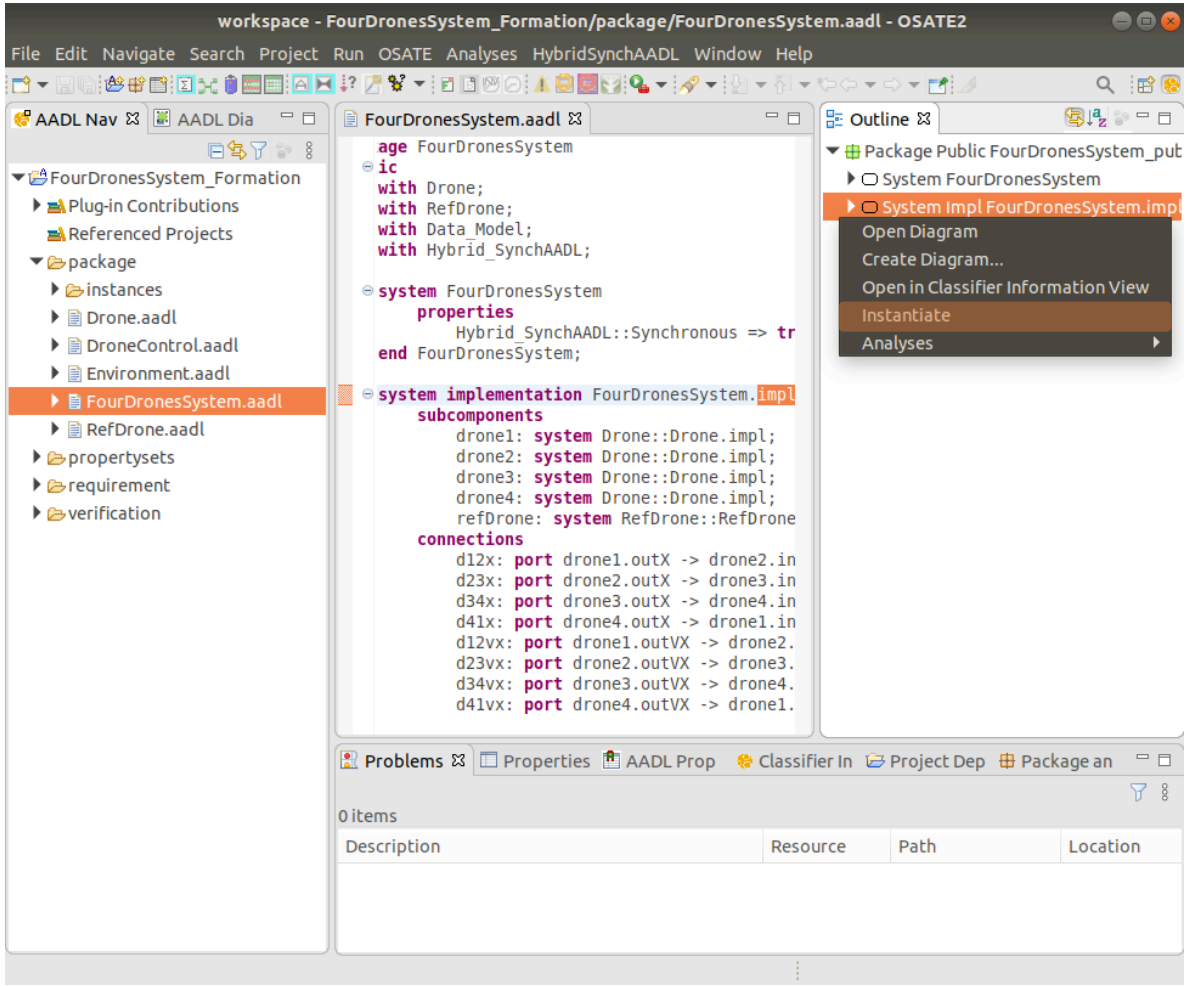
- `FourDroneSystem.aadl` contains the top-level system component.



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`.

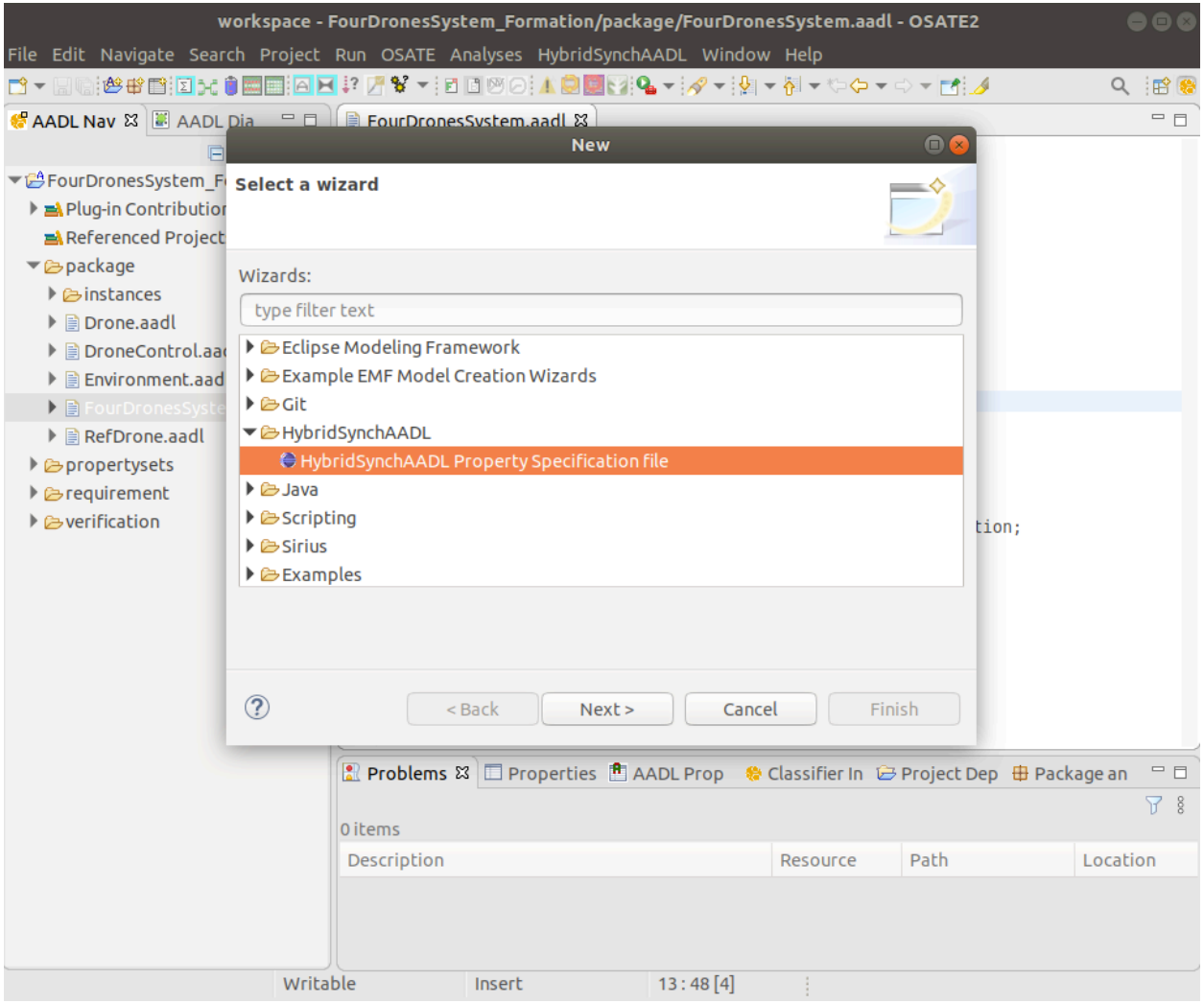


Outline

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

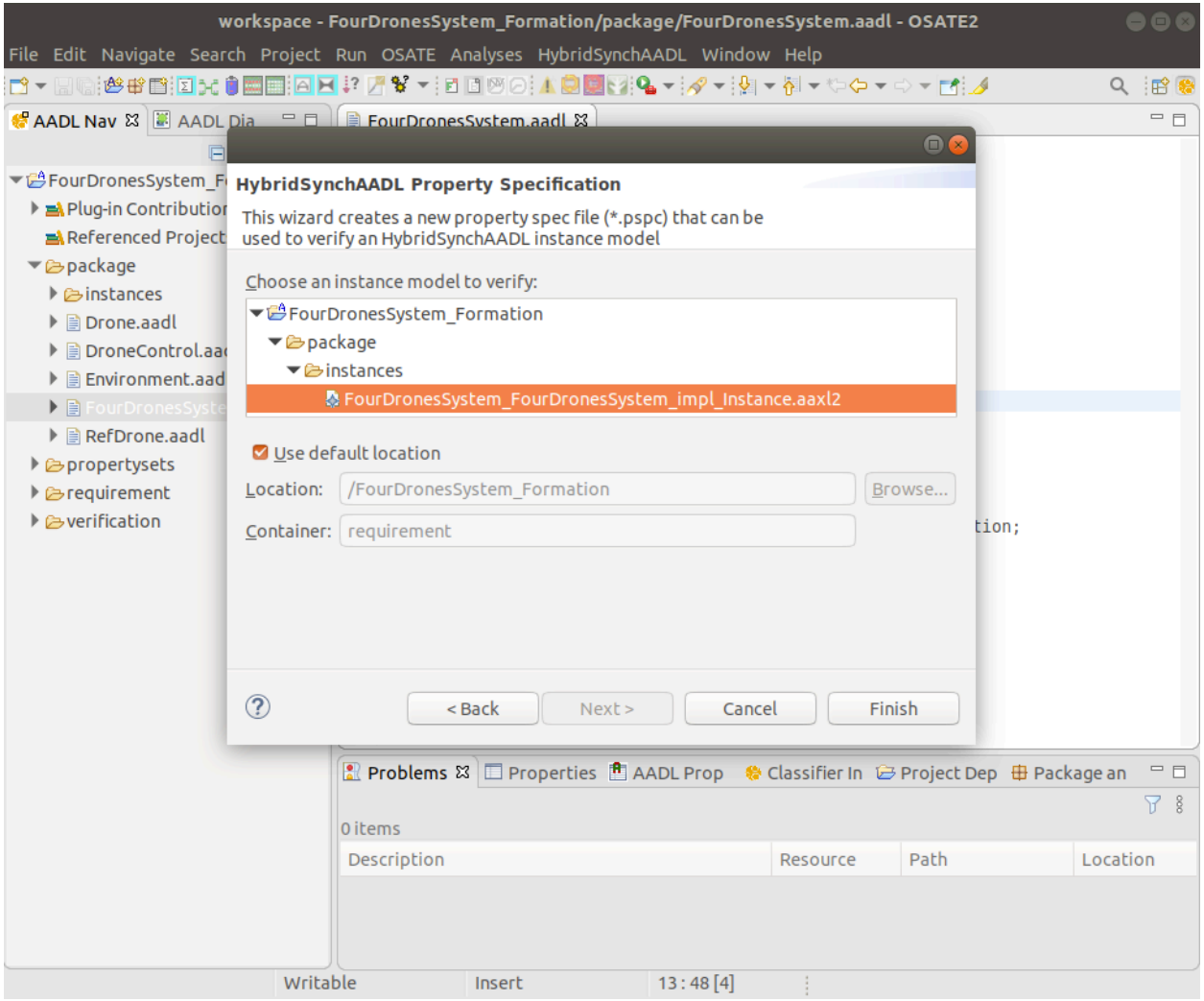
Creating PSPC Files

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



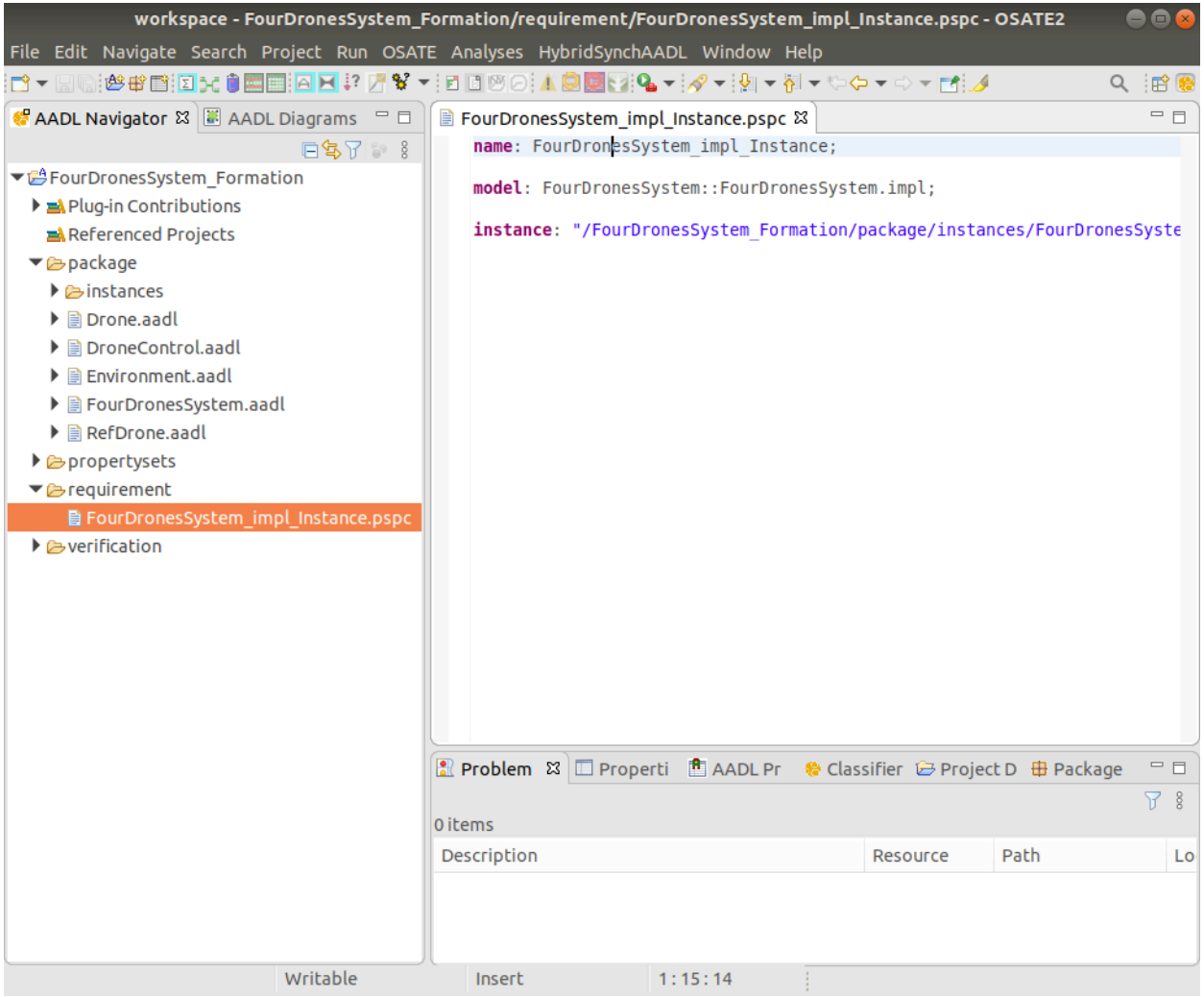
Creating PSPC Files

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



Creating PSPC Files

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

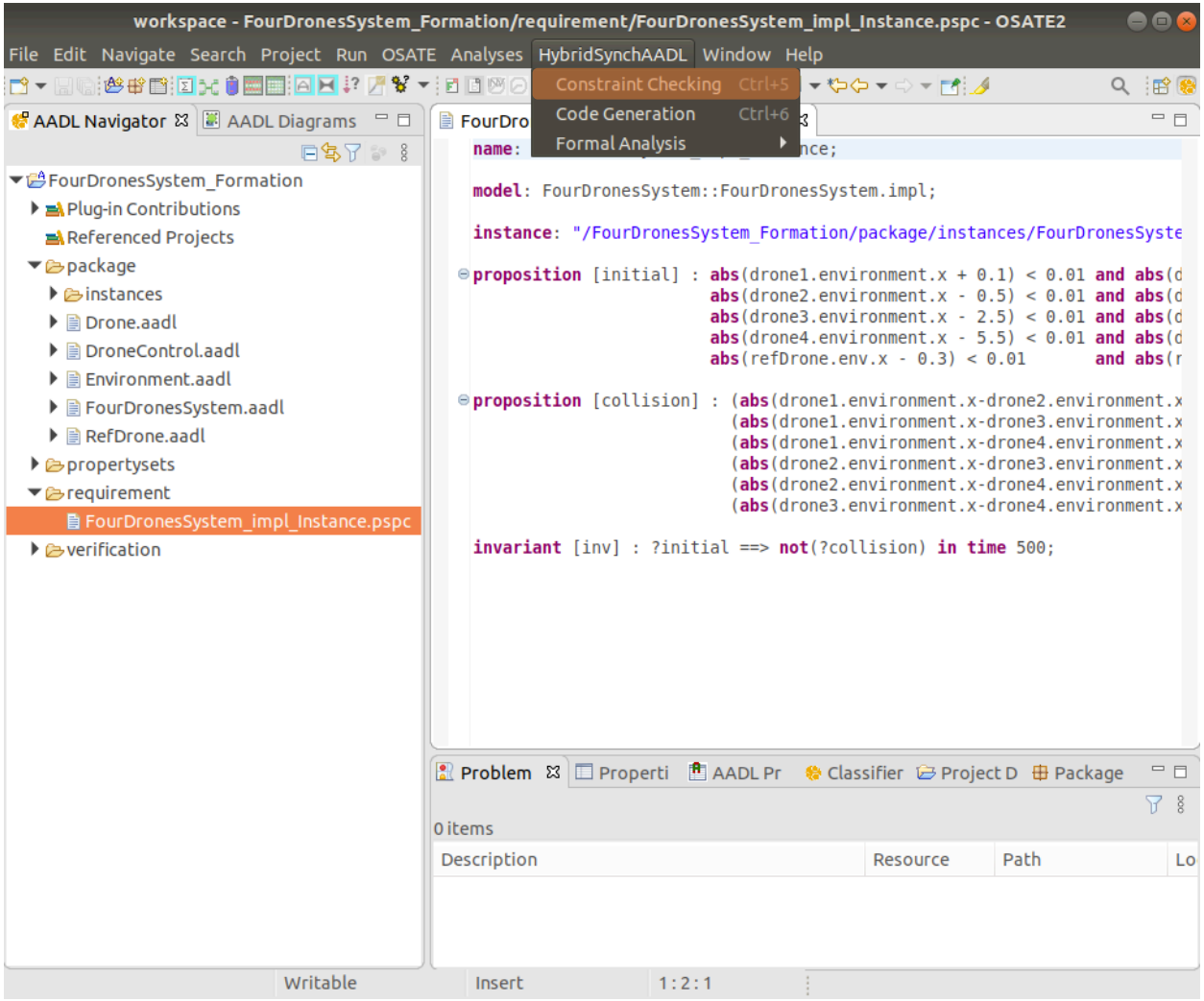


Outline

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

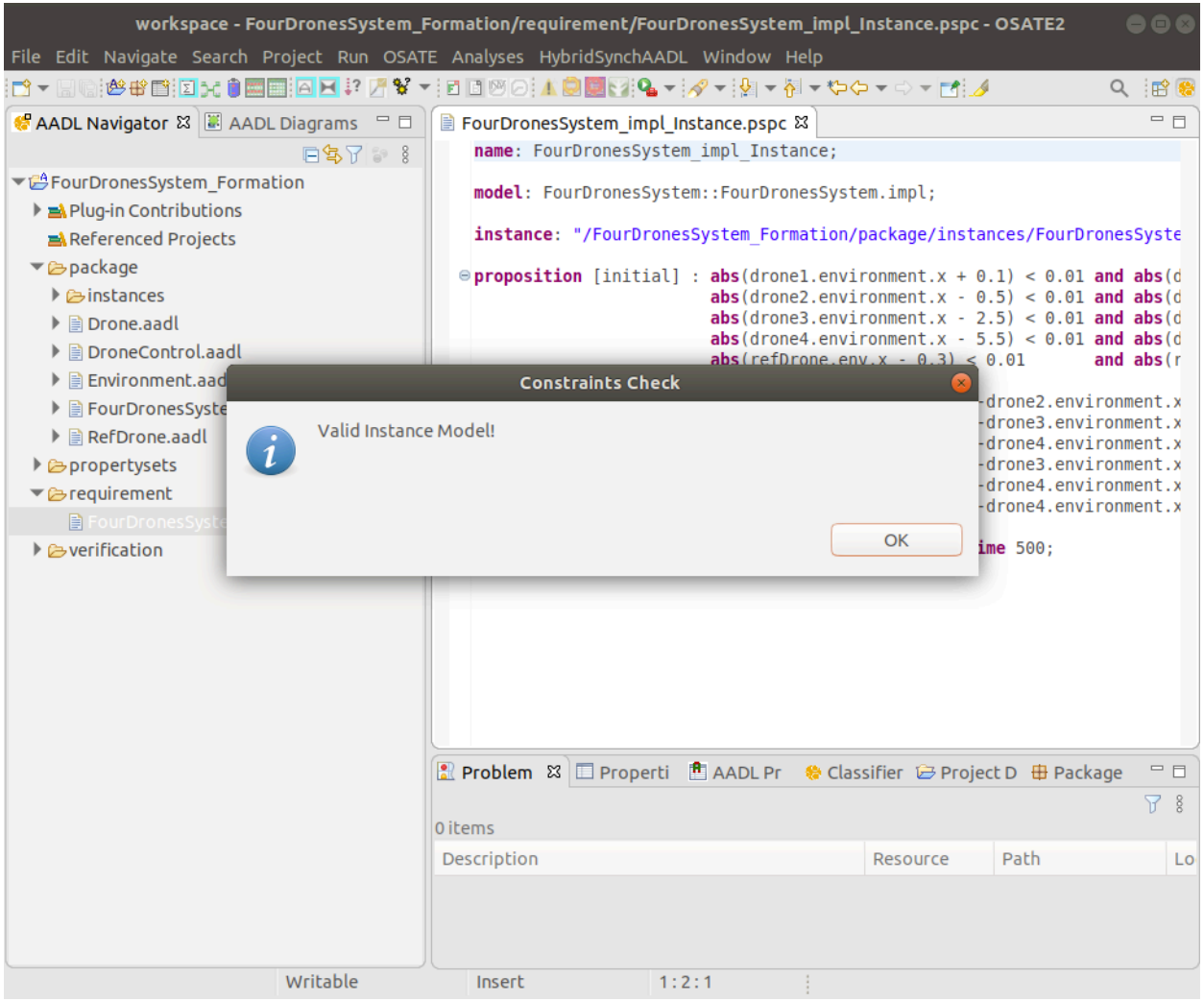
Checking HybridSynchAADL Constraints

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



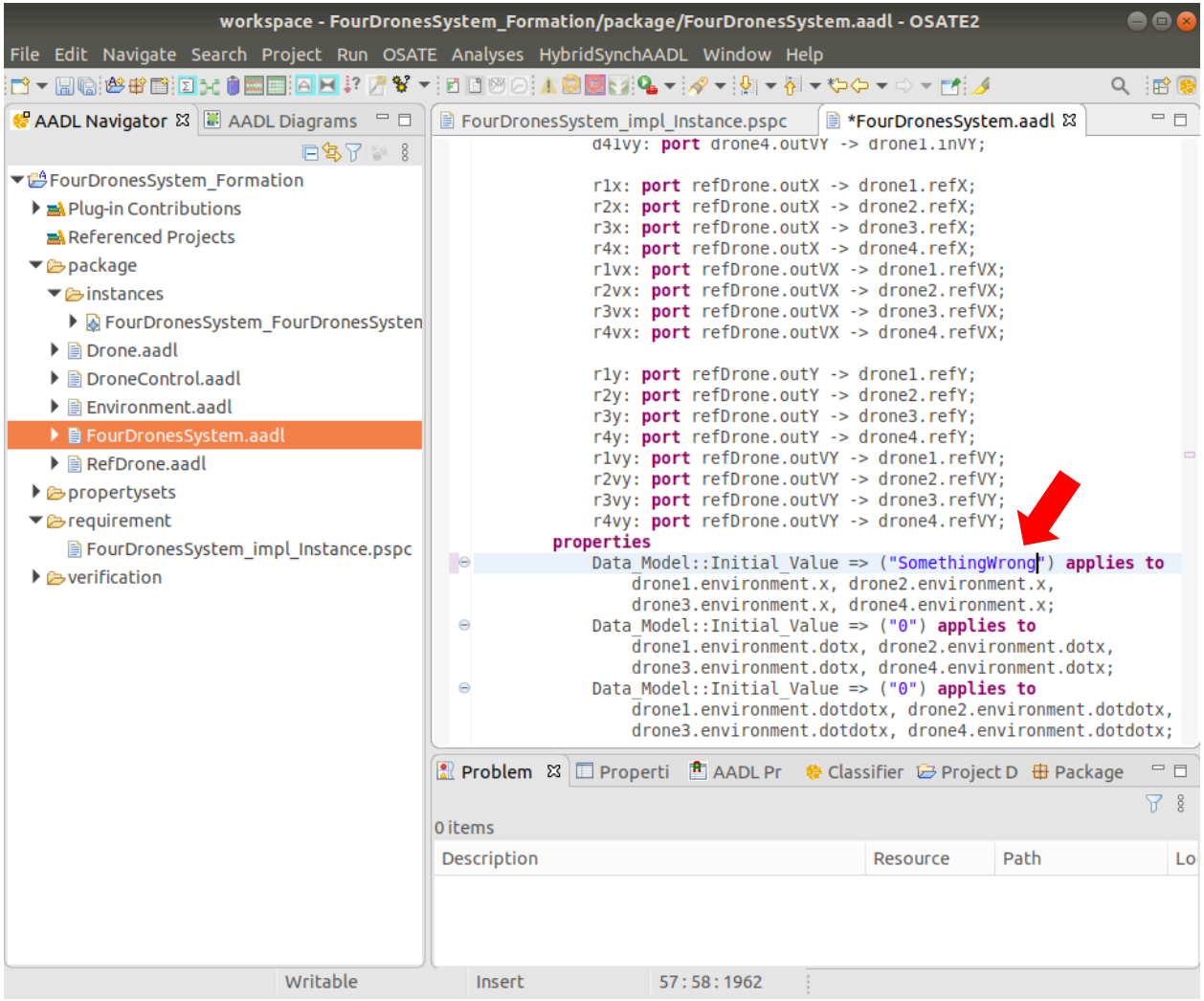
Checking HybridSynchAADL Constraints

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



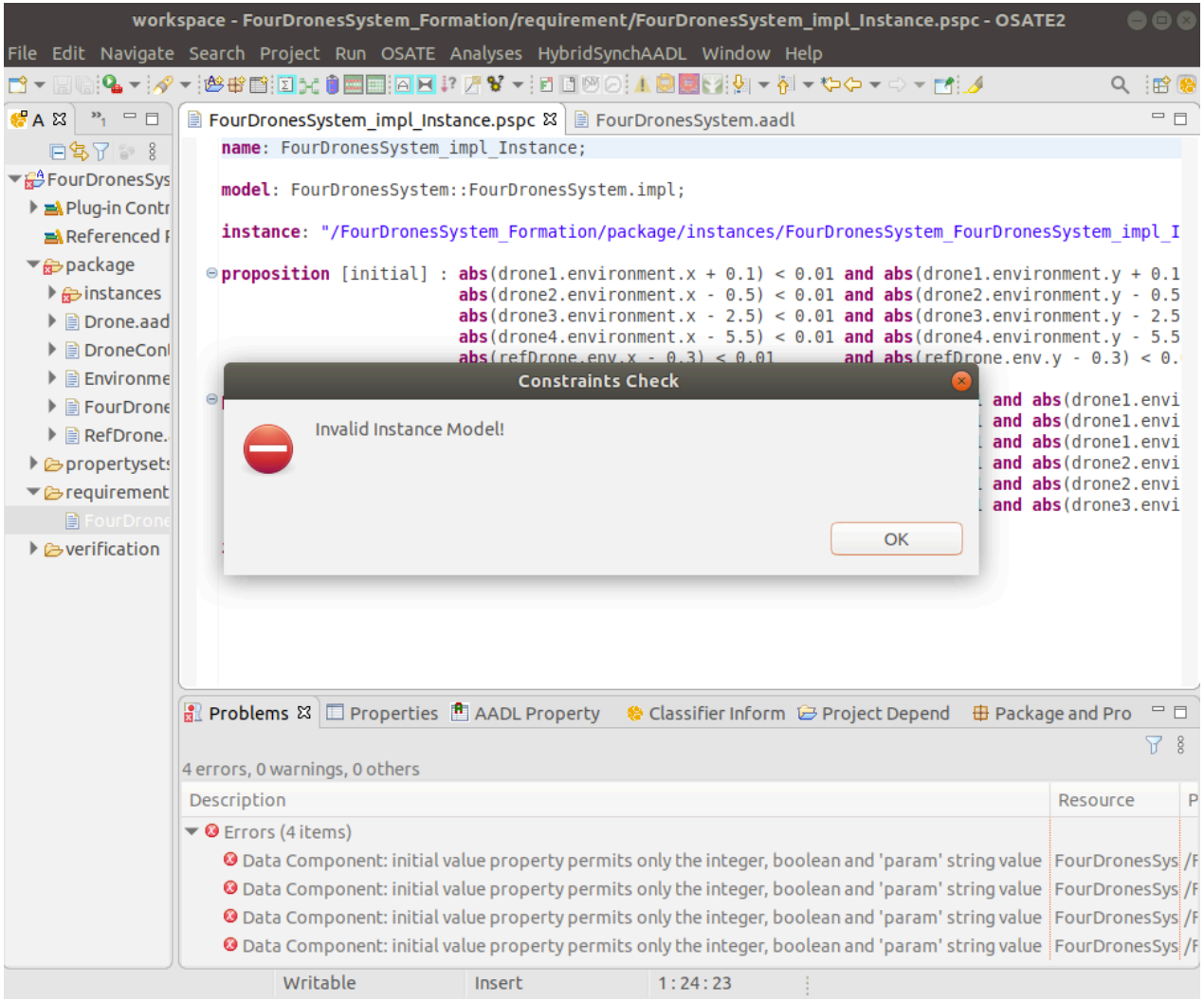
Constraints Check – Erroneous Model

- 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.`



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.

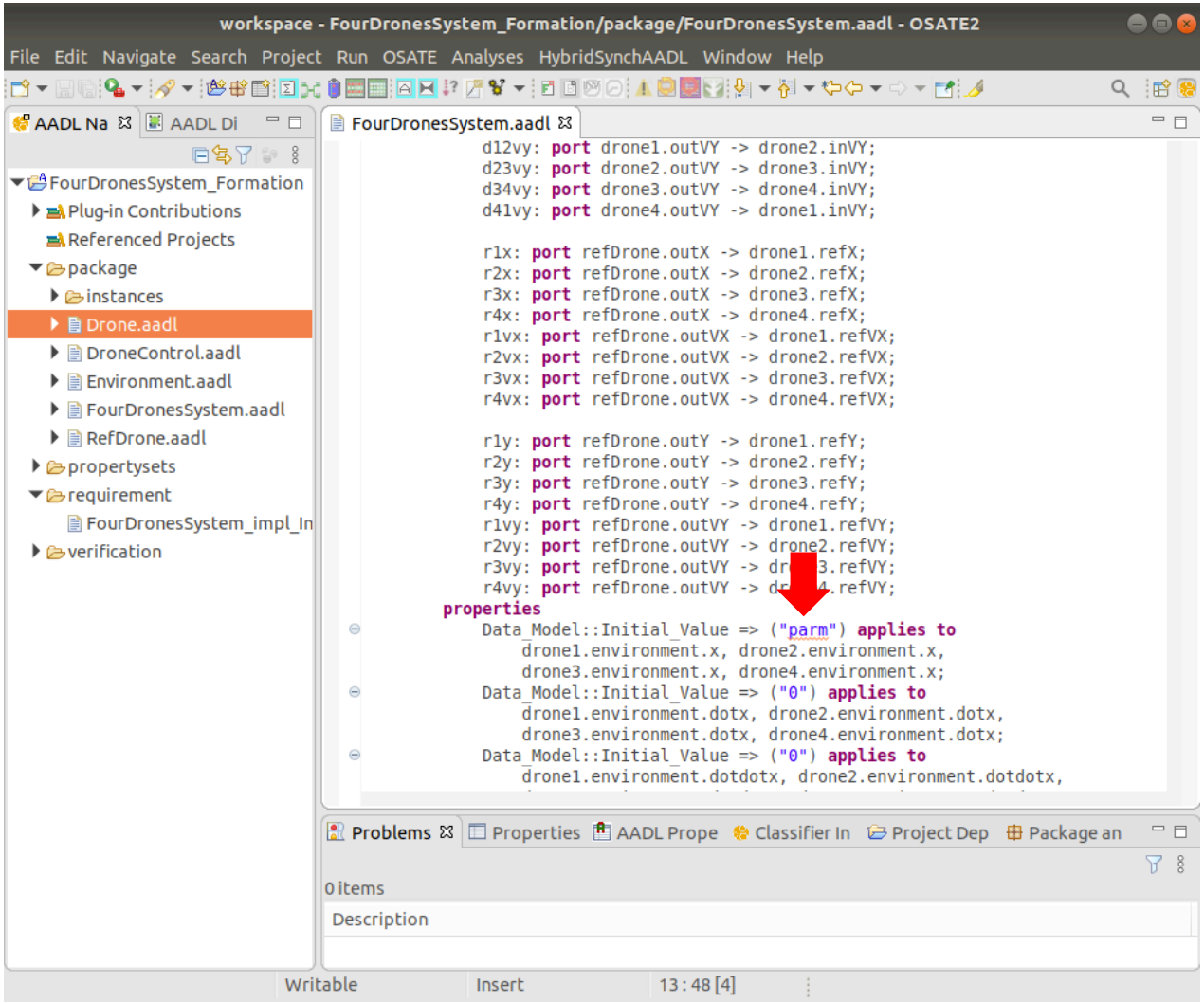


Outline

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

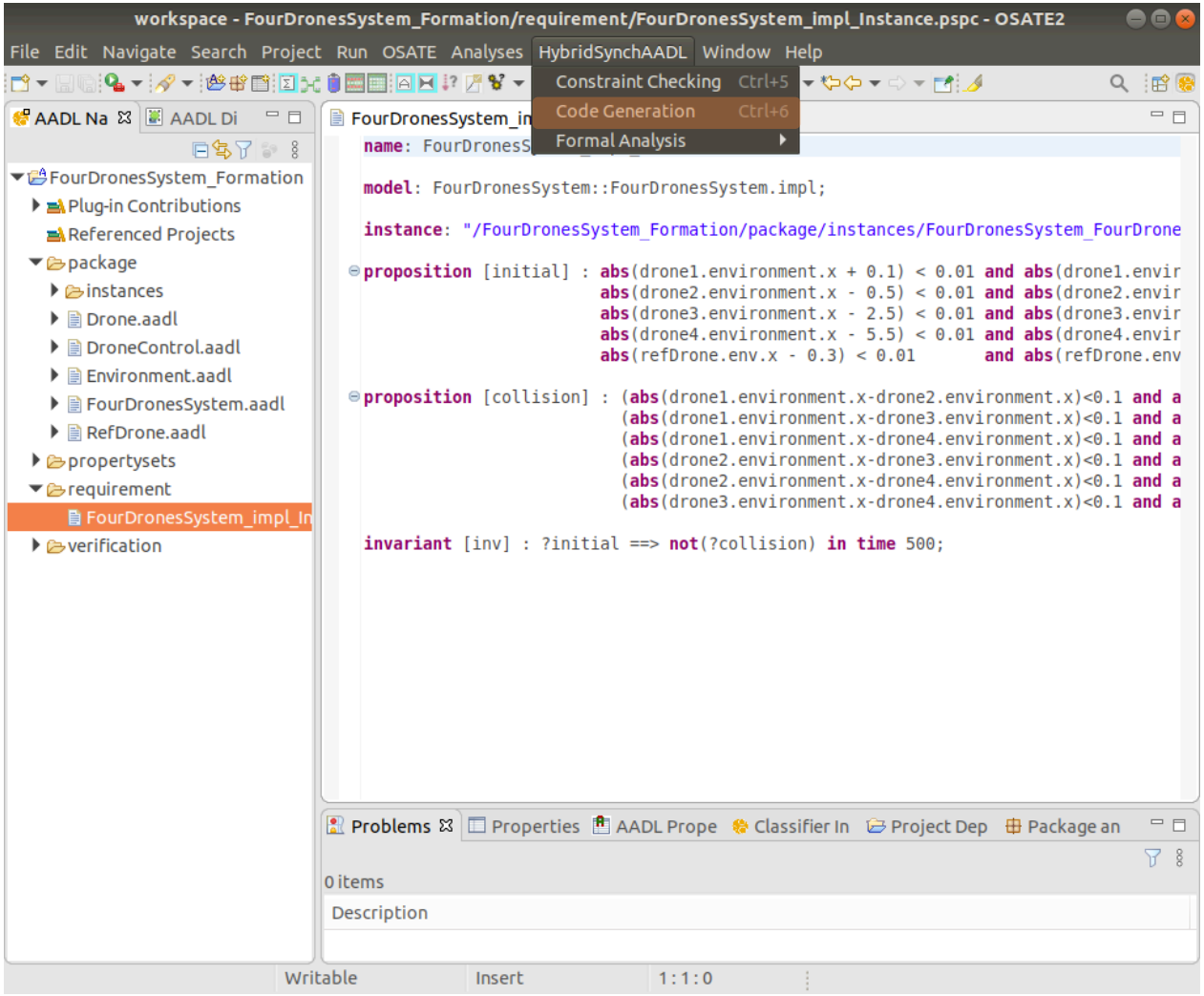
The FourDronesSystem Example

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



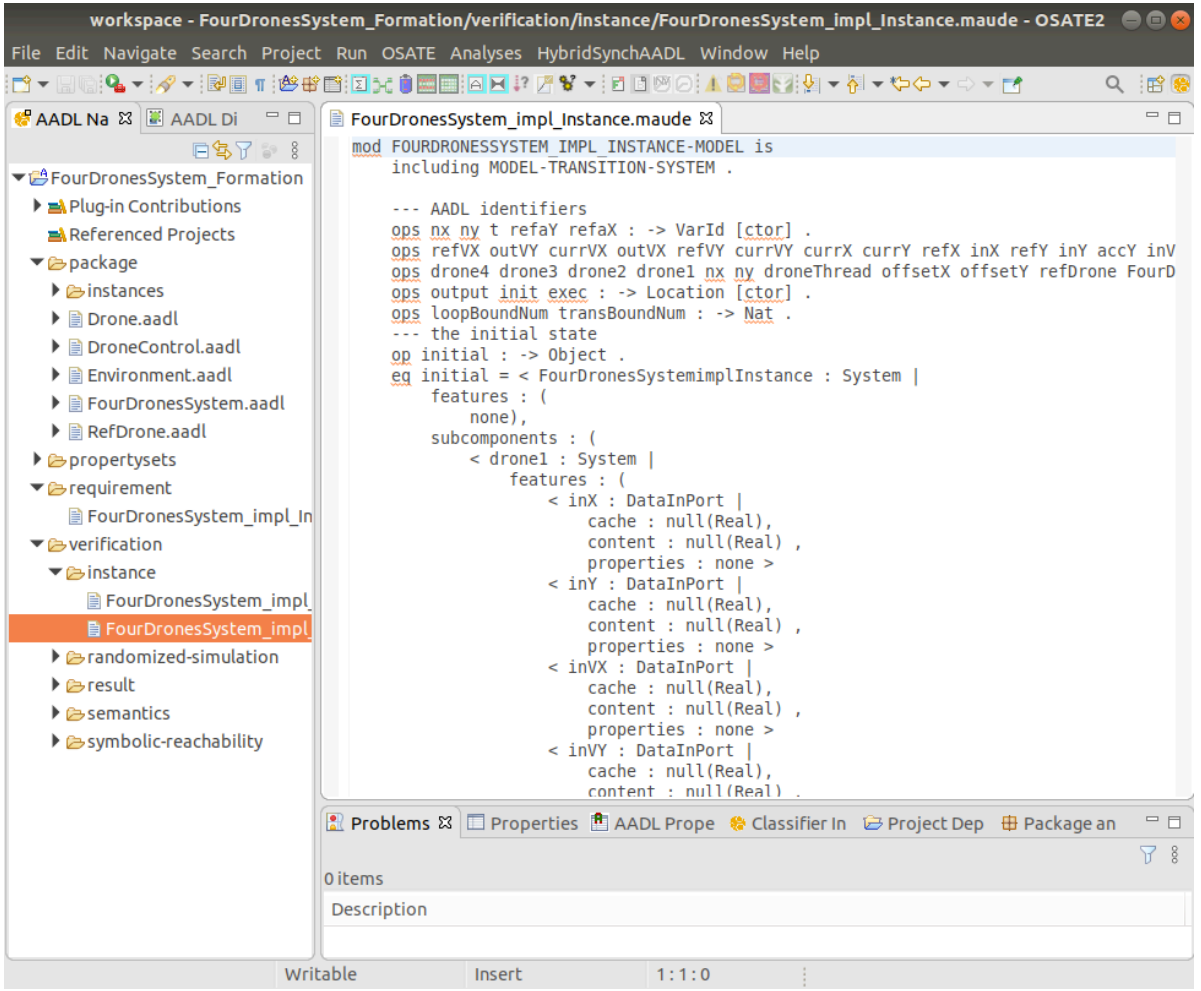
Maude Code Generation

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



Maude Code Generation

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

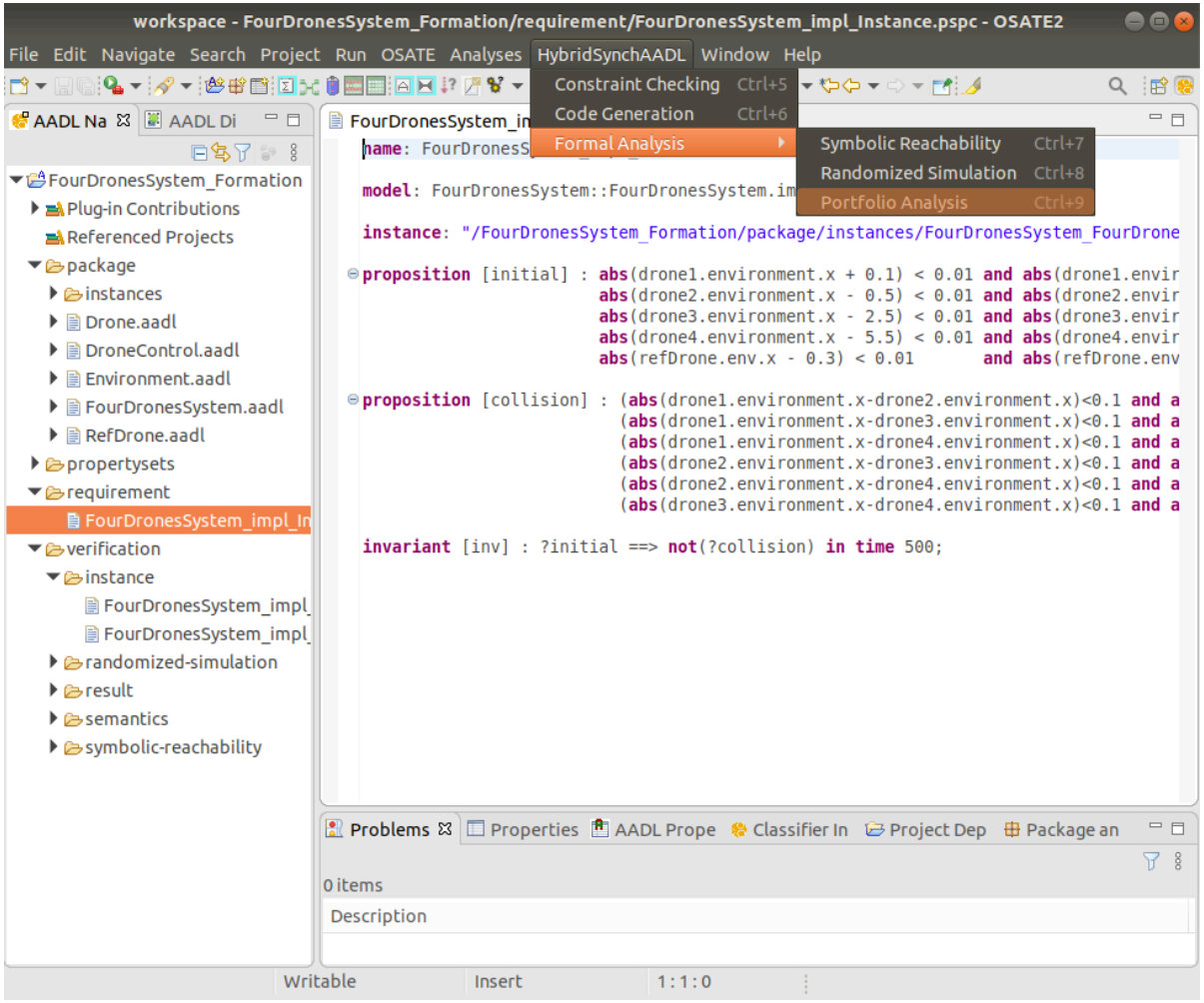


Outline

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

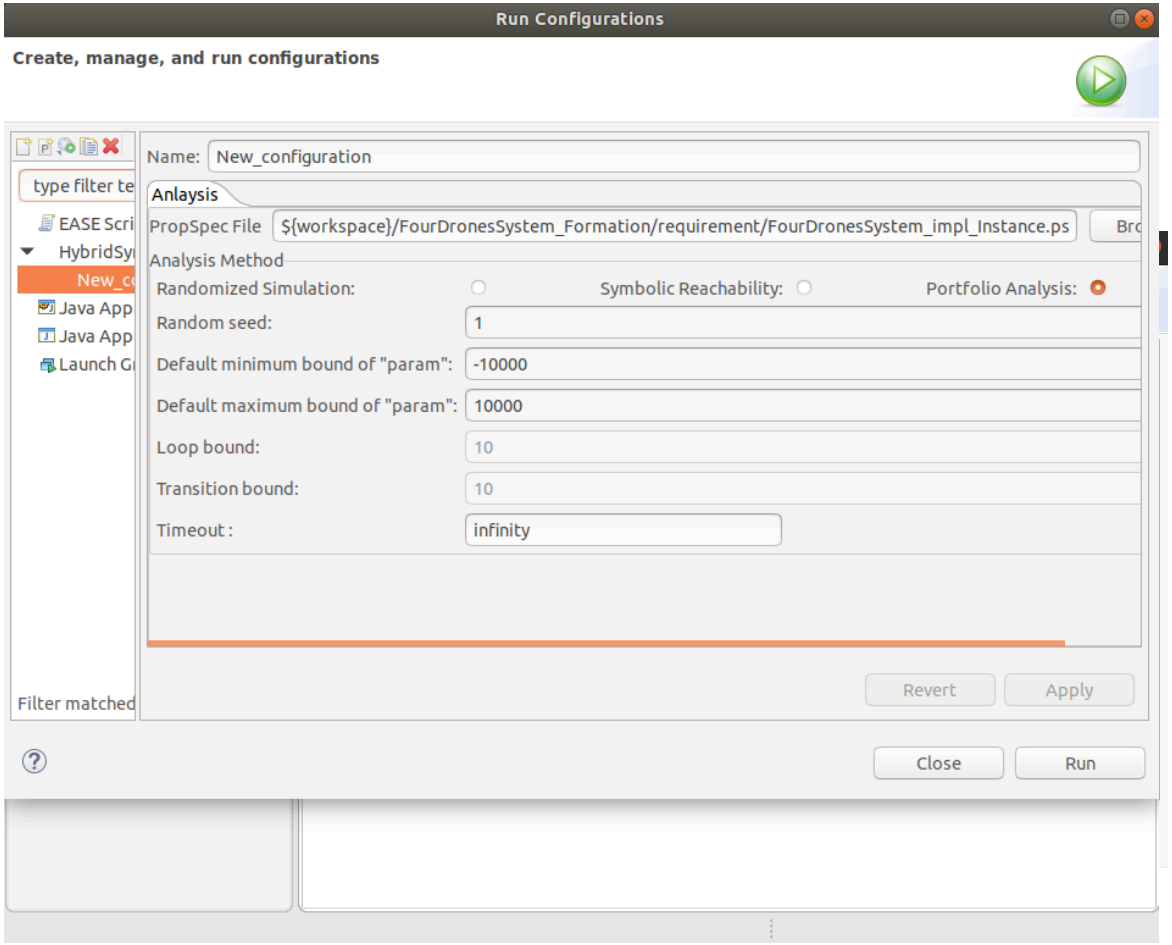
Portfolio Analysis

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



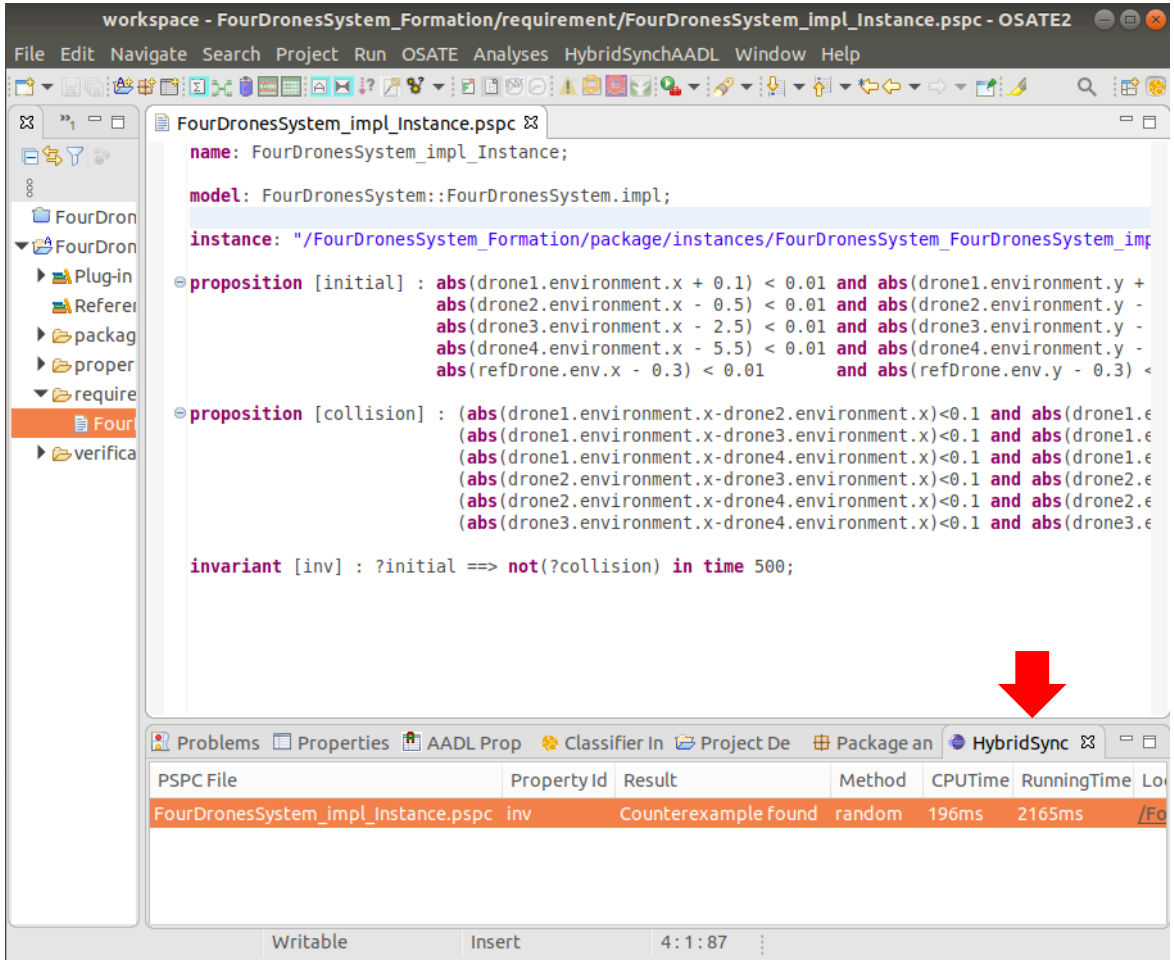
Portfolio Analysis

- 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.



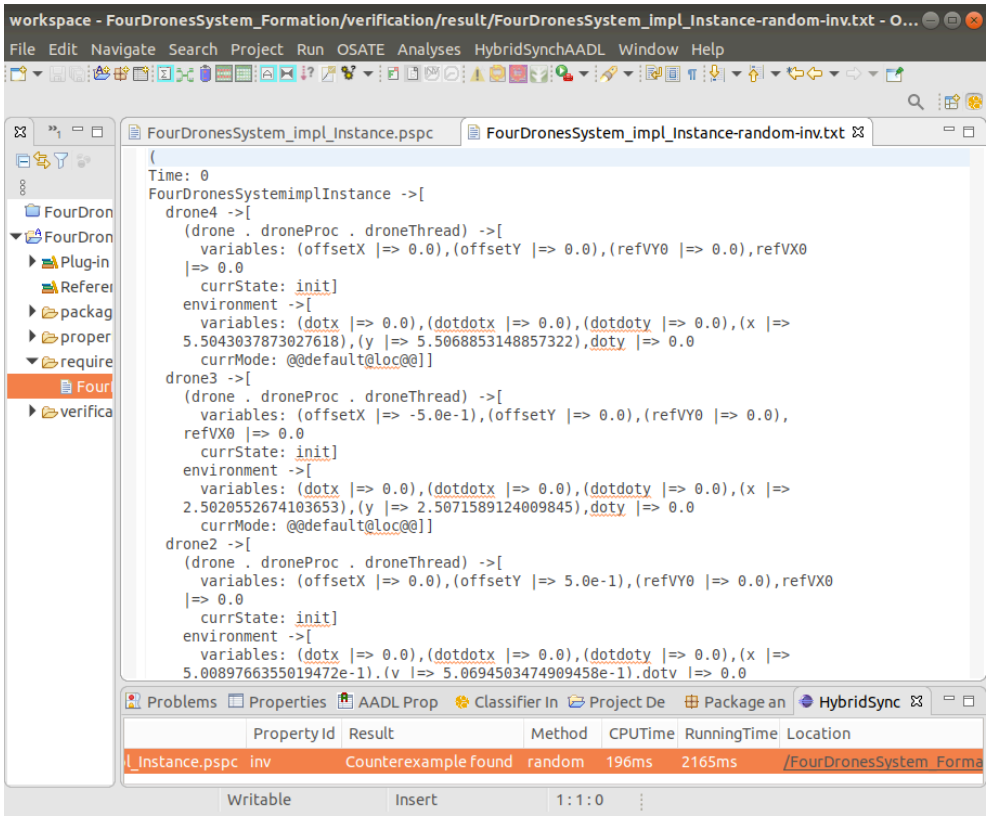
Analysis Results

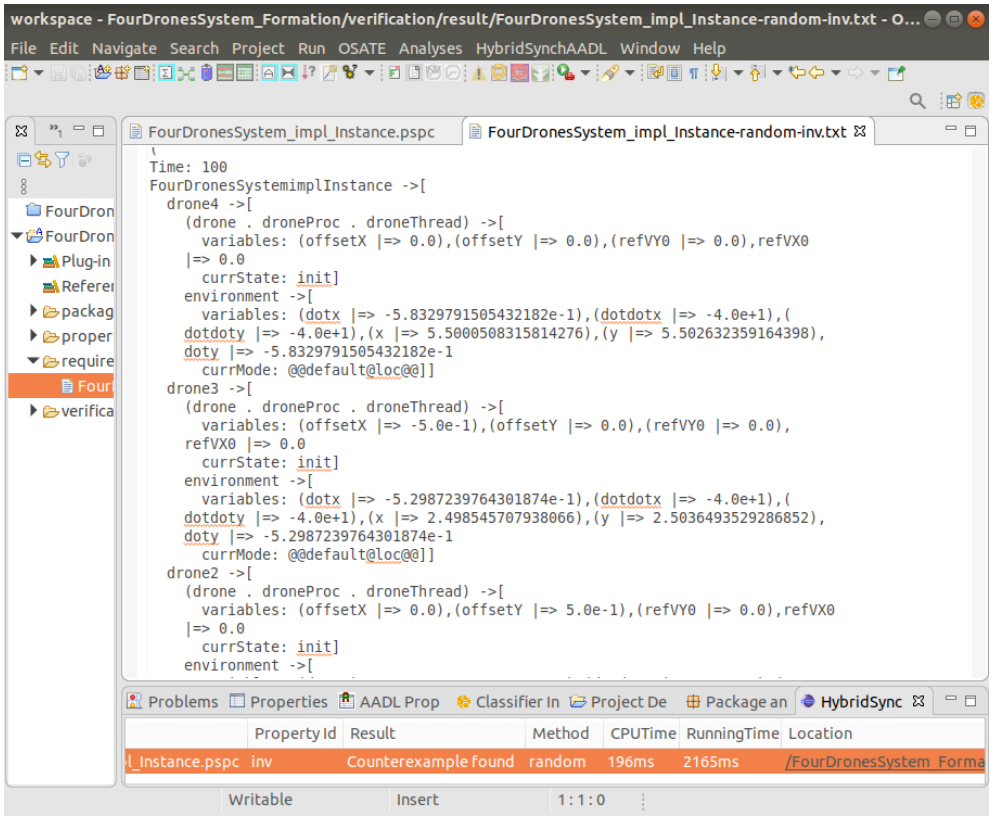
- The HybridSynchAADL Result view shows the analysis results.



Counterexample

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





Thank you!