

Documentation for repository **ViewVerification**

Created by Fabian Kahlert
2018 - May - 22th

Purpose

- Checks whether EmbeddedMontiArc models fulfill all properties defined by EmbeddedMontiView models (= views)
- EmbeddedMontiView:
 - Textual structured diagram language to abstractly describe properties of C&C architectures
 - Based on EmbeddedMontiArc with additional features:
 - Keywords for “atomic”-ness and “interface-complete”-ness
 - Untyped and anonymous ports (e.g. `port in ? ?;`)
 - Textual anonymous ports (e.g. `port in ? $port1;`)
 - Abstract connectors & effectors
 - Chains of connectors
 - Effectors include influence of computations

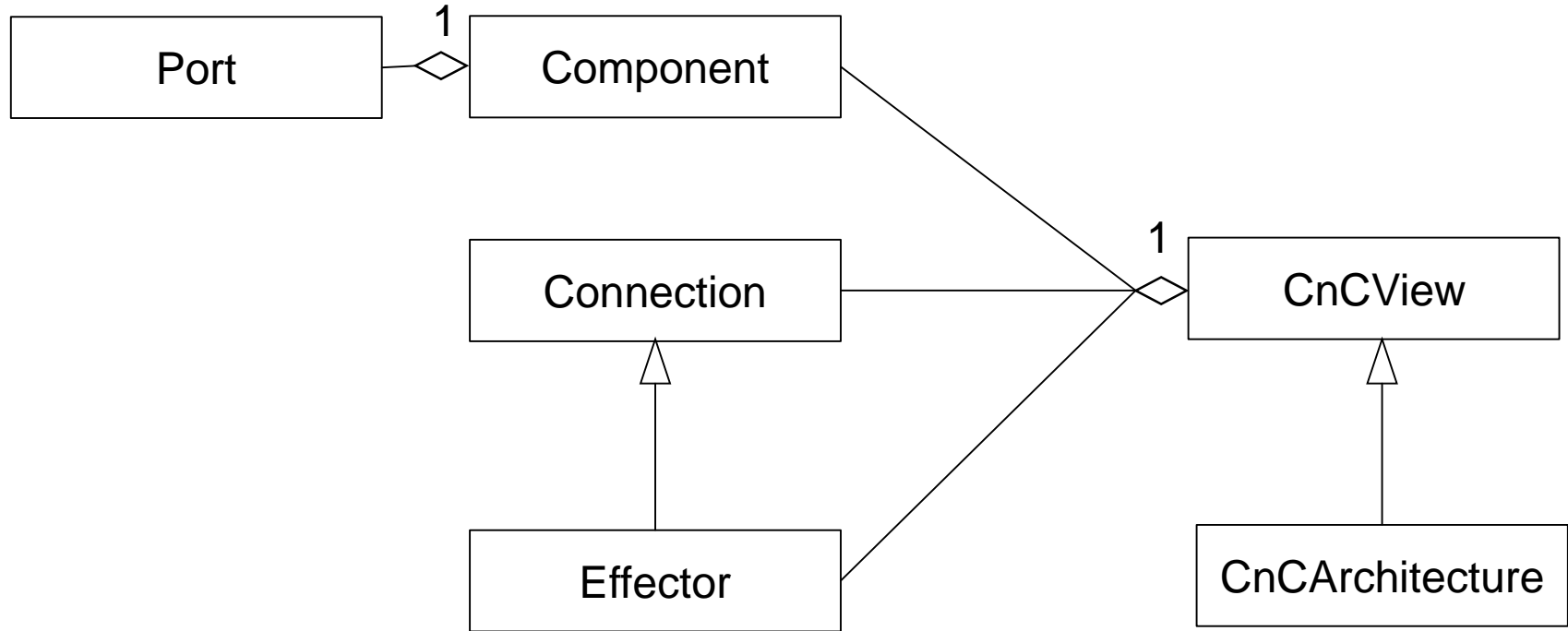
Packages and Functionality

Important classes only

Package de.rwth.cnc.model

- Classes
 - CnCArchitecture
 - CnCView
 - Component
 - Port
 - Connection
 - Effector
- Internal representation of all needed informations about models and views

Class Relations



Package de.rwth.cnc.viewverification

- EmbeddedMontiArcLoader
 - Loading of C&C models, returns internal representation
- EmbeddedMontiViewLoader
 - Loading of views
 - Converting from internal representation to textual model (e.g. for witnesses)
- ViewVerifier
 - Verifies models against views
 - Returns inconsistencies and witnesses

Package de.rwth.cnc.viewverification.checks

- Each class checks whether the model satisfies a specific property of the view
 - Check for ...
 - missing components
 - interface mismatches
 - hierarchy mismatches
 - missing connectors
 - missing effectors
 - not atomic components
 - interface-completeness violations

Package de.rwth.cnc.viewverification.inconsistency

- Inconsistency
 - InconsistencyHierarchyMismatch
 - InconsistencyInterfaceMismatch
 - InconsistencyMissingComponent
 - InconsistencyMissingConnection
 - etc.
- Provide important information about the specific errors
- InconsistenciesData
 - Contains lists of all inconsistencies
- InconsistencyItem
 - Represents an existing inconsistency
 - Contains a natural language description of the error

Package `de.rwth.cnc.viewverification.witness`

- `WitnessGenerator`
 - Generates multiple negative witnesses or a positive witness using the following classes
- `GeneratePositiveWitnessView`
 - Generates a positive witness as a `CnCView` showing the specific implementation of the view's elements
- `GenerateInconsistencyView`
 - Contains methods for generation of witnesses for each kind of inconsistency
- `GenerateInconsistencyDesc`
 - Generates natural language descriptions which are given as comment at the beginning of each view
- `WitnessGeneratorHelper`
 - Contains some helping methods needed in the upper classes

Tests

- EmbeddedMontiArcLoaderTest
- EmbeddedMontiViewLoaderTest
 - Test whether specific models can and are loaded correctly
- AvionicsSystemExampleTest
- BumperBotEmergencyTest
- PumpingSystemExampleTest
- RoboticArmTest
 - Test several industrial examples and their requirement views
- CustomerAcceptanceTest
 - Additionally tests rare cases, also in combination with special units
- OCLVerificationTest
 - Additionally tests cases with OCL expressions

Software Quality

- Code Quality (Codeclimate): D
- Test Coverage: 85%