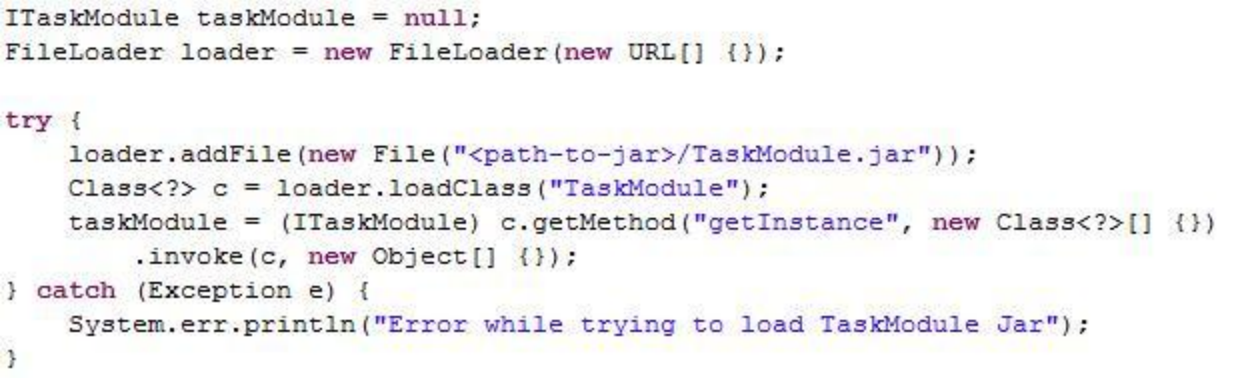
Task Module API

Developers: Assaf Israel & Eli Nazarov

Working Flow - Setup

- ITaskModule, IExperiment & IDataModel will be available via Standards.jar
- Concrete implementation will be available via a the TaskModule.Jar
- Dynamically load jar file to ITaskModule (almost like games)



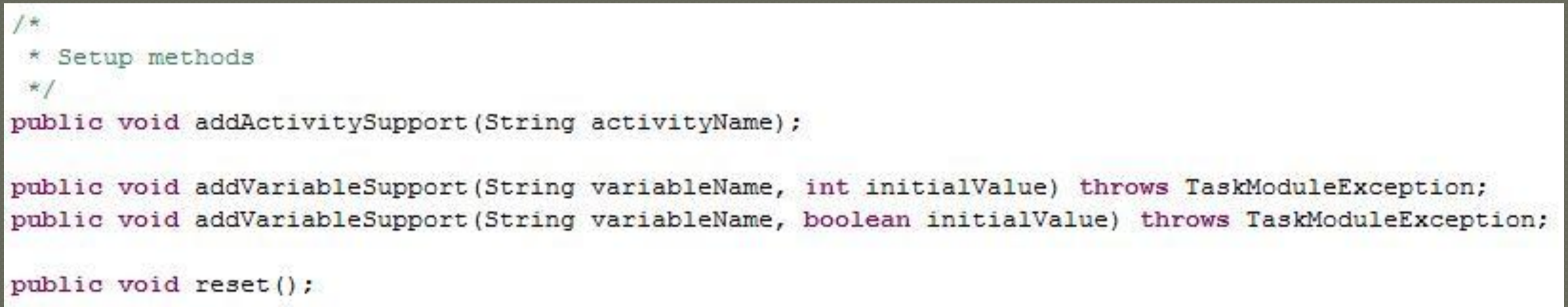
```
ITaskModule taskModule = null;
FileLoader loader = new FileLoader(new URL[] {});

try {
    loader.addFile(new File("<path-to-jar>/TaskModule.jar"));
    Class<?> c = loader.loadClass("TaskModule");
    taskModule = (ITaskModule) c.getMethod("getInstance", new Class<?>[] {})
        .invoke(c, new Object[] {});
} catch (Exception e) {
    System.err.println("Error while trying to load TaskModule Jar");
}
```

Working Flow - Overview

- You know your program best
  - Methods/Variables names
  - Some methods may not exists
  - Implementation independent
- Call it what you want, and let us know when it happens

Working Flow - Initialization

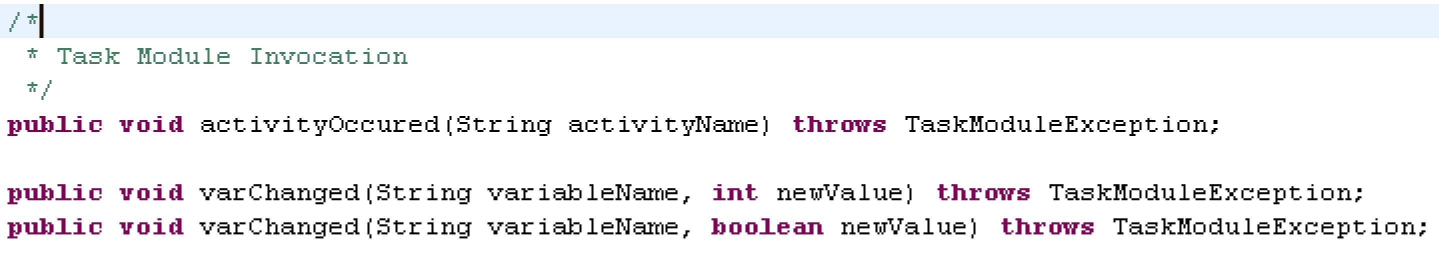
- Declare and name your “Activities”
- Declare and name your Variables
  - Currently we’re supporting Boolean & Integer var.



```
/*
 * Setup methods
 */
public void addActivitySupport(String activityName);

public void addVariableSupport(String variableName, int initialValue) throws TaskModuleException;
public void addVariableSupport(String variableName, boolean initialValue) throws TaskModuleException;

public void reset();
```

Working Flow – Invocation

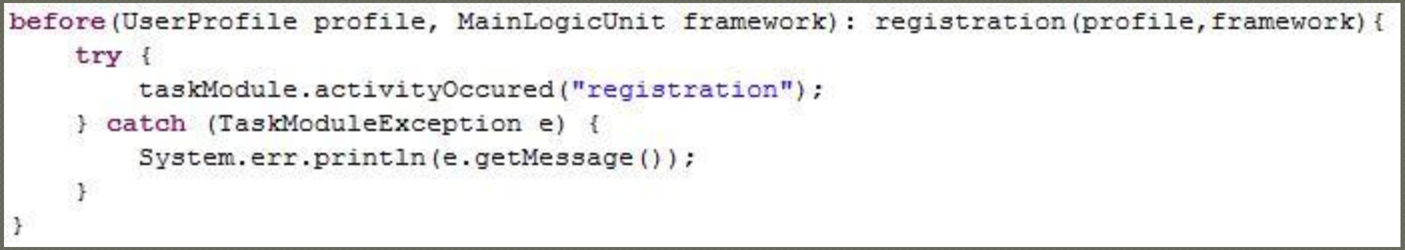
- You are responsible for letting us know when Activities occur & when Variables change.



```
/*
 * Task Module Invocation
 */
public void activityOccured(String activityName) throws TaskModuleException;

public void varChanged(String variableName, int newValue) throws TaskModuleException;
public void varChanged(String variableName, boolean newValue) throws TaskModuleException;
```

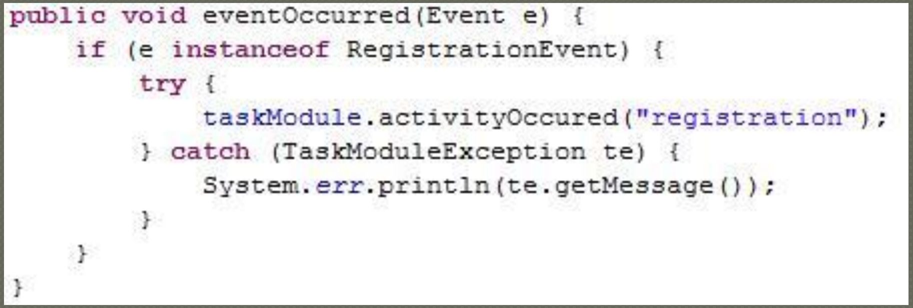
Working Flow – Invocation

- AspectJ



```
before(UserProfile profile, MainLogicUnit framework): registration(profile,framework){
    try {
        taskModule.activityOccured("registration");
    } catch (TaskModuleException e) {
        System.err.println(e.getMessage());
    }
}
```

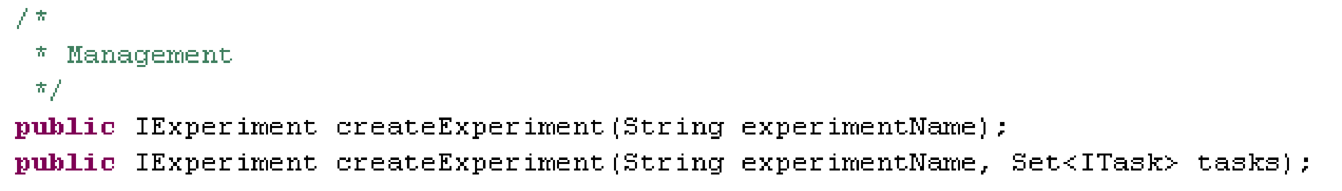
- Events



```
public void eventOccurred(Event e) {
    if (e instanceof RegistrationEvent) {
        try {
            taskModule.activityOccured("registration");
        } catch (TaskModuleException te) {
            System.err.println(te.getMessage());
        }
    }
}
```

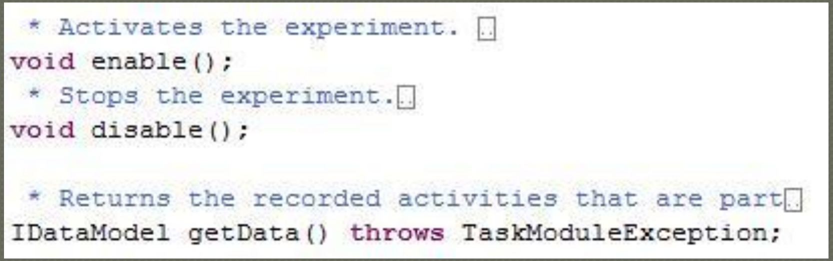
Working Flow – Management

- Tasks are compiled ConGlog scripts
- Experiments are smart Task containers
  - Serves as a Set of Tasks



```
/*
 * Management
 */
public IExperiment createExperiment(String experimentName);
public IExperiment createExperiment(String experimentName, Set<ITask> tasks);
```

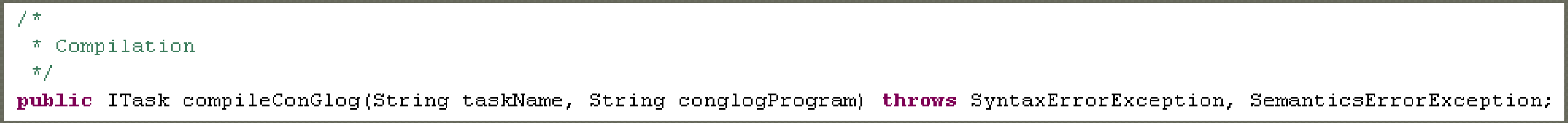
  - You can Enable/Disable Experiment recording



```
 * Activates the experiment. □
void enable();
 * Stops the experiment.□
void disable();

 * Returns the recorded activities that are part□
IDataModel getData() throws TaskModuleException;
```

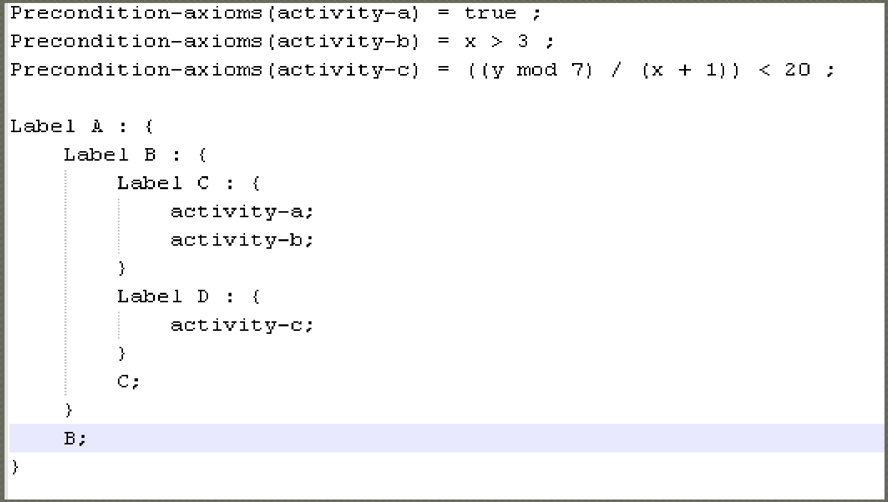
Working Flow – Compiler

- Task Creation
  - ConGlog scripts are compiled to Tasks

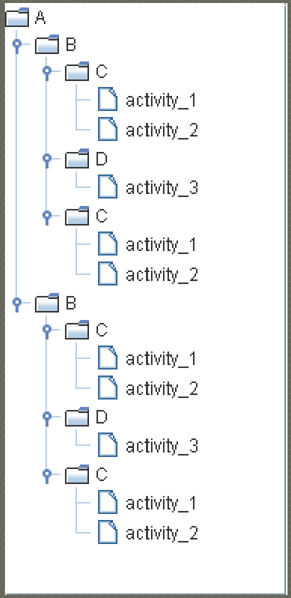


```
/*
 * Compilation
 */
public ITask compileConGlog(String taskName, String conglogProgram) throws SyntaxErrorException, SemanticsErrorException;
```

- ConGlog Script



```
Precondition-axioms(activity-a) = true ;
Precondition-axioms(activity-b) = x > 3 ;
Precondition-axioms(activity-c) = ((y mod 7) / (x + 1)) < 20 ;

Label A : {
    Label B : {
        Label C : {
            activity-a;
            activity-b;
        }
        Label D : {
            activity-c;
        }
        C;
    }
    B;
}
```



```
A
  B
    C
      activity_1
      activity_2
    D
      activity_3
    C
      activity_1
      activity_2
  B
    C
      activity_1
      activity_2
    D
      activity_3
    C
      activity_1
      activity_2
```

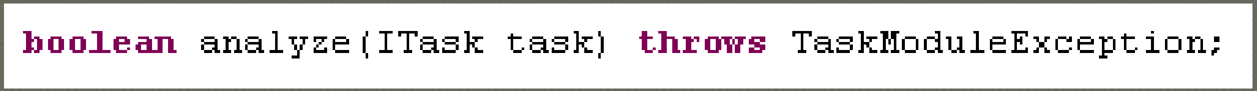
Working Flow – Data

- Experiment Management
  - Experiment recording can be enabled/disabled (via IExperiment interface)

- Data Model
  - Generic storage of data.
  - Can be exported to CSV file.
  - Experiment scope.
  - Task Module scope.

Working Flow – Analyzer

- Default Analyzer
  (via IDataModel interface)



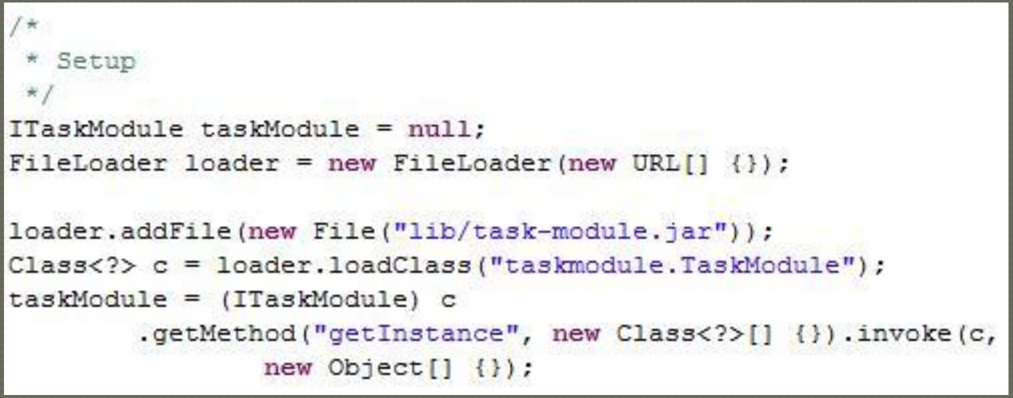
```
boolean analyze(ITask task) throws TaskModuleException;
```

E2E Example

- The following example is an end to end simple scenario implementation using AspectJ

E2E Example - Setup

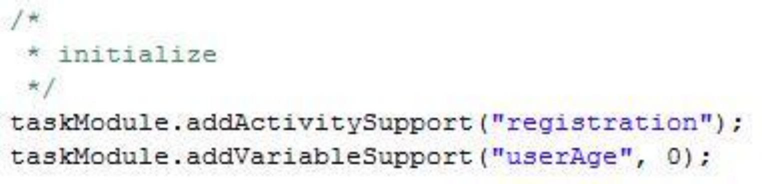
- task-module.jar has been put in lib folder
- load the concrete “taskmodule.TaskModule” class
- Interfaces are available via Standards.jar
- ITaskModule is design as a singleton, so use “getInstance”



```
/*
 * Setup
 */
ITaskModule taskModule = null;
FileLoader loader = new FileLoader(new URL[] {});

loader.addFile(new File("lib/task-module.jar"));
Class<?> c = loader.loadClass("taskmodule.TaskModule");
taskModule = (ITaskModule) c
        .getMethod("getInstance", new Class<?>[] {}).invoke(c,
                new Object[] {});
```

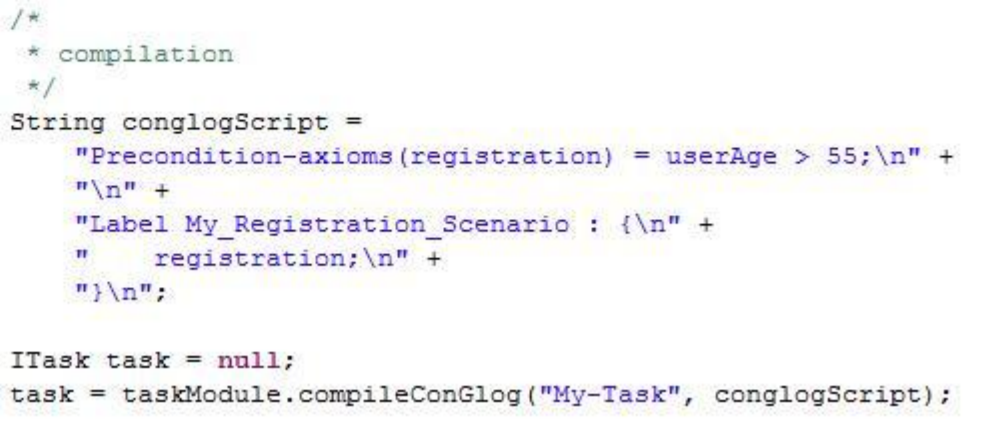
E2E Example – Initialize & Compile

- We’ll demonstrate a simple registration scenario



```
/*
 * initialize
 */
taskModule.addActivitySupport("registration");
taskModule.addVariableSupport("userAge", 0);
```

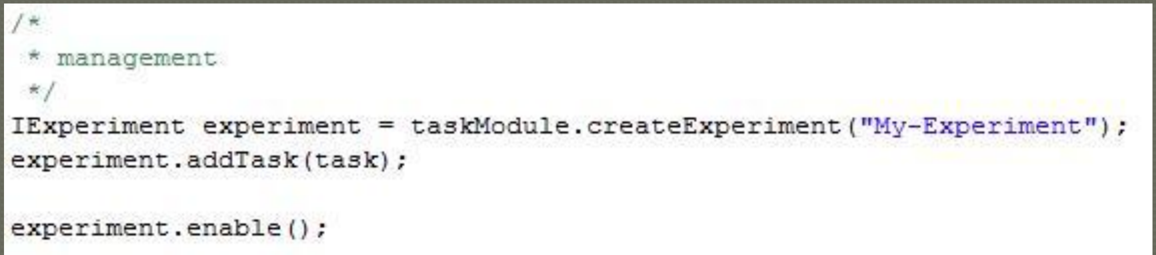
- The following script, checks if the user age is above 55



```
/*
 * compilation
 */
String conglogScript =
    "Precondition-axioms(registration) = userAge > 55;\n" +
    "\n" +
    "Label My_Registration_Scenario : {\n" +
    "     registration;\n" +
    "}\n";

ITask task = null;
task = taskModule.compileConGlog("My-Task", conglogScript);
```

E2E Example – Experiment

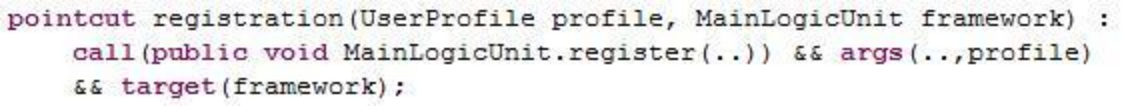
- We’ll create a new experiment with the compiled task and enable the recording for it.



```
/*
 * management
 */
IExperiment experiment = taskModule.createExperiment("My-Experiment");
experiment.addTask(task);

experiment.enable();
```

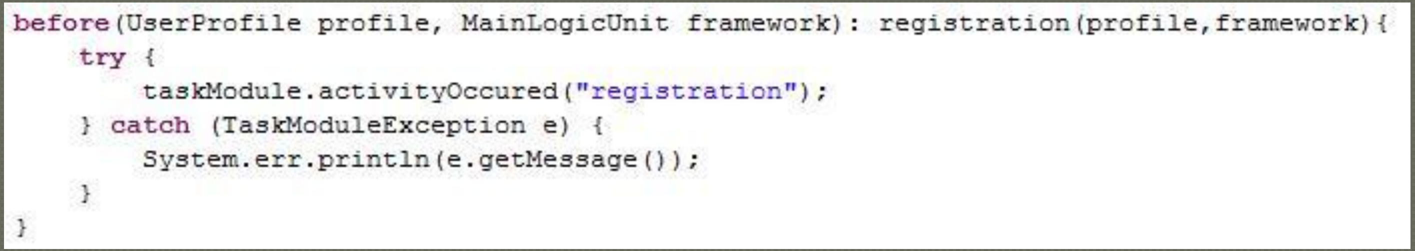
E2E Example – AspectJ

- Registration pointcut



```
pointcut registration(UserProfile profile, MainLogicUnit framework) :
    call(public void MainLogicUnit.register(..)) && args(..,profile)
    && target(framework);
```

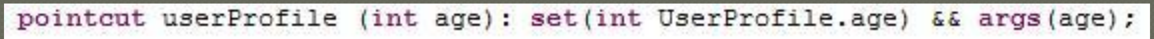
- Registration advice



```
before(UserProfile profile, MainLogicUnit framework): registration(profile,framework){
    try {
        taskModule.activityOccured("registration");
    } catch (TaskModuleException e) {
        System.err.println(e.getMessage());
    }
}
```

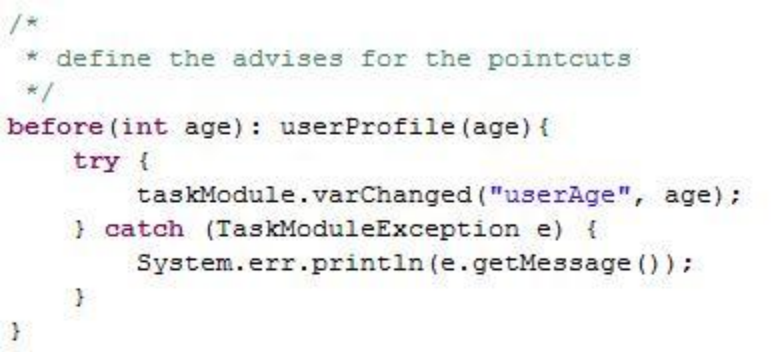
E2E Example – AspectJ Cont.

- User age pointcut



```
pointcut userProfile (int age): set(int UserProfile.age) && args(age);
```

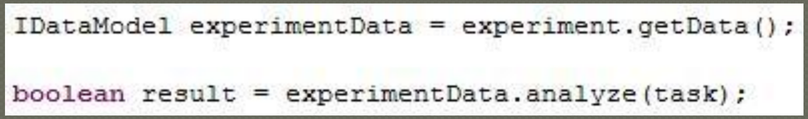
- User age advice



```
/*
 * define the advises for the pointcuts
 */
before(int age): userProfile(age){
    try {
        taskModule.varChanged("userAge", age);
    } catch (TaskModuleException e) {
        System.err.println(e.getMessage());
    }
}
```

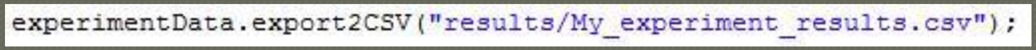
E2E Example – Analyze

- After we’ve played a bit with the system, we’re ready to get the results



```
IDataModel experimentData = experiment.getData();

boolean result = experimentData.analyze(task);
```

- Export them to a file



```
experimentData.export2CSV("results/My_experiment_results.csv");
```