MATSim software developments

Kai Nagel

June 23, 2018

Outline

1. Getting started
   - MATSim example project, MATSim code examples
   - MATSim versions and maven
2. MATSim scripts in Java
   - Scripts-in-Java
   - Code examples
   - Dependency injection
3. MATSim GUI
4. Regression tests
   - Travis tests for github projects
5. Conclusion

MATSim example project

`https://github.com/matsim-org` → matsim-example-project

Get this from github into your own programming environment so that you are able to use it (for beginners it is easiest to first fork the project; for experts there are several options).

Works *without* cloning "matsim" repository

One does not need to clone the github repository called "matsim"

Instead, pulls MATSim "from the cloud" (technically: from a maven repo).

→ in general, remove "matsim" from your IDE ...
... (e.g. "close project" in eclipse) ...
... or don't even clone it in the first place.

Can still browse all MATSim source code through the maven dependency.

MATSim code examples

`https://github.com/matsim-org` → matsim-code-examples

We try to keep *class names* stable, but package names/directory structure still change. So search material via class name

- use `https://github.com/matsim-org/matsim-code-examples/find/0.10.x`

or

- in IDE after cloning it to there,

Outline

1. Getting started
   - MATSim example project, MATSim code examples
   - MATSim versions and maven
2. MATSim scripts in Java
   - Scripts-in-Java
   - Code examples
   - Dependency injection
3. MATSim GUI
4. Regression tests
   - Travis tests for github projects
5. Conclusion

MATSim versions, maven

The MATSim version is in the pom.xml (see matsim-example-project):

```
    <dependency>
        <groupId>org.matsim</groupId>
        <artifactId>matsim</artifactId>
        <version>0.9.0</version>
    </dependency>
```

Also the responsible maven repository is entered in the pom.xml:

```
    <repository>
        <id>matsim</id>
        <url>https://dl.bintray.com/matsim/matsim</url>
    </repository>
```

Check the pom.xml in matsim-code-examples for variants (snapshots, monthly snapshots, ...).

Outline

1. Getting started
   - MATSim example project, MATSim code examples
   - MATSim versions and maven
2. MATSim scripts in Java
   - Scripts-in-Java
   - Code examples
   - Dependency injection
3. MATSim GUI
4. Regression tests
   - Travis tests for github projects
5. Conclusion

The (currently) favourite MATSim script

```
Config config = ConfigUtils.createConfig() ;
// or ... = ConfigUtils.loadConfig(filename) ;

// now modify config if you want/need

// ---

Scenario scenario = ScenarioUtils.createScenario( config ) ;
// or ... = ScenarioUtils.loadScenario( config ) ;

// now modify scenario if you want/need

// ---

Controler controler = new Controler( scenario ) ;

// now modify controler if you want/need

// ---

controler.run() ;
```

Outline

1. Getting started
   - MATSim example project, MATSim code examples
   - MATSim versions and maven
2. MATSim scripts in Java
   - Scripts-in-Java
   - Code examples
   - Dependency injection
3. MATSim GUI
4. Regression tests
   - Travis tests for github projects
5. Conclusion

Stable code examples

Now in separate repository: `https://github.com/matsim-org` → matsim-code-examples.

Default branch is a version (i.e. not `master`). → table links in FAQ etc.

This is, even for a released MATSim version, a "live" repository (e.g. in reaction to questions).

Quick look at directory/package structure.

Outline

1. Getting started
   - MATSim example project, MATSim code examples
   - MATSim versions and maven
2. MATSim scripts in Java
   - Scripts-in-Java
   - Code examples
   - Dependency injection
3. MATSim GUI
4. Regression tests
   - Travis tests for github projects
5. Conclusion

Controler addOverridingModule syntax

[[addOverridingModule ... show in IDE]]

[[note import correct AbstractModule (from matsim)]]

Controler addOverridingModule syntax, ctd

```
controler.addOverridingModule(new AbstractModule(){
   @Override public void install() {
      this.addControlerListenerBinding()... ;
      this.addEventHandlerBinding()... ;
      this.addMobsimListenerBinding()... ;

      this.bindScoringFunctionFactory()... ;
      this.bindMobsim()... ;
      this.bindLeastCostPathCalculatorFactory()... ;

      this.addPlanStrategyBinding(selectorName)... ;
      this.addTravelDisutilityFactoryBinding(mode)... ;
      this.addTravelTimeBinding(mode)... ;
      this.addPlanSelectorForRemovalBinding(selectorName)... ;
      // ---
      this.bindNetworkTravelTime()... ;
      this.carTravelDisutilityFactoryKey()... ;
   }
});
```

Bind to instances

Easiest version.

```
XxxImpl xxx = new XxxImpl(...) ;
...
this.bind(XxxInterface.class).to(xxx) ;
```

with

```
class XxxImpl implements XxxInterface {
   ...
}
```

Bind to classes

Often best compromise between simplicity and flexibility.

```
this.bind(XxxInterface.class).to(XxxImpl.class) ;
```

with

```
class XxxImpl implements XxxInterface {
   ...
   @Inject
   XxxImpl( Arg1 arg1, Arg2 arg2, ... ) {
      ...
   }
   ...
}
```

Note constructor never explicitly called (and can make it package-protected so that nobody is tempted to do so).

Everything that is bound somewhere can be injected here.

Bind to providers

When binding to classes is no longer powerful enough.

In MATSim often also often for historical reasons.

Binding/inject syntax, advantages

- Can change dependencies of module without having to change upstream code (no explicit constructor call, see above).
- Framework sorts out dependencies automatically.

MATSim extensions

Preferred way to run extensions (e.g. a contrib):

1. Add dependency in `pom.xml`.
2. Find code example (either in `matsim-code-examples`, or in the extension's repository.
3. Copy the relevant code snippets into your own script-in-java.

How to find code examples → matsim-code-examples

…

Also some examples for extensions!

How to find code examples → ControlerDefaultsModule

...

Remember this!

Once one has this as a starting point, it becomes relatively clear (I think).

Without this, it is a bit difficult to find (I think).

How to find injected classes?

See log file.

MATSim GUI from MATSim example project

MATSim GUI part of exampe project.

Can run from IDE.

Can call your own script-in-java.

(Most) filenames are now relative to location of config.

Run `mvn package` and have clickable jar file.

→ process to package specialized MATSim jars for self and others.

Outline

1. Getting started
   - MATSim example project, MATSim code examples
   - MATSim versions and maven
2. MATSim scripts in Java
   - Scripts-in-Java
   - Code examples
   - Dependency injection
3. MATSim GUI
4. Regression tests
   - Travis tests for github projects
5. Conclusion

Automatic tests: Travis

If you have forked the project, setting up automatic travis regression tests is very easy (just log into travis with your github account and follow instructions).

matsim-example-project has very simple `.travis.yml` file.

matsim-code-examples has more elaborate `.travis.yml` file.

Tests go into `src/test/java`.

Conclusion

1 Fork `matsim-example-project` on github.

2 Register with travis to obtain automatic regression tests.

3 Pull into your IDE. Maven will sort out dependencies. No need to download the MATSim main repository.

4 Makes it easy to switch MATSim versions (in `pom.xml`).

5 Write MATSim scripts-in-Java.

6 Use `matsim-code-examples` for coding examples.

7 Also see there for examples of how to use extensions.

8 Remember `ControlerDefaultsModule`.