

MATSim advanced class

Kai Nagel

November 2, 2020

OUTLINE

- 1 Introduction
- 2 Getting started
- 3 MATSim scripts in Java
- 4 Guice in MATSim
- 5 Regression tests
- 6 Conclusion (of this part)
- 7 Running on the math cluster and other remote compute engines
- 8 Plans innovation, plans removal, diversity
- 9 Train vs road evaluation in BVWP
- 10 Move on to java13
- 11 Emissions contrib
- 12 Visualization using VIA
- 13 Dashboards etc.: Towards automatic results presentation?
- 14 Visualization and dashboards using lavascript

Purpose of this class

VSP teaching tracks:

- How to use transport simulations. (Mod.Sim.V, Ana.Bew.V.Sys)
- Telematics methods. (Meth.V.Telem)
- How to write (plugins to) transport simulations. (OOP, MATSim)

Present class: Continuation of “how to write” track.

Purpose of this class

VSP teaching tracks:

- How to use transport simulations. (Mod.Sim.V, Ana.Bew.V.Sys)
- Telematics methods. (Meth.V.Telem)
- How to write (plugins to) transport simulations. (OOP, MATSim)

Present class: Continuation of “how to write” track.

Purpose

- Go beyond “basic” MATSim class.
- “Internal” training (lots of elements that were added/modified over last 2 years or so).
- (Hopefully) improve MATSim documentation.

Approach

- Bring laptop, try things out while we go.
- (Hopefully) improve documentation on the way.
- Please bring own topics.
- Please feed back.
- Universitas magistrorum et scholarium – “Gemeinschaft der Lehrenden und Lernenden”

Legalese

- Who needs/wants a certificate?

OUTLINE

- 1 Introduction
- 2 Getting started**
- 3 MATSim scripts in Java
- 4 Guice in MATSim
- 5 Regression tests
- 6 Conclusion (of this part)
- 7 Running on the math cluster and other remote compute engines
- 8 Plans innovation, plans removal, diversity
- 9 Train vs road evaluation in BVWP
- 10 Move on to java13
- 11 Emissions contrib
- 12 Visualization using VIA
- 13 Dashboards etc.: Towards automatic results presentation?
- 14 Visualization and dashboards using lavascript

Outline

2 Getting started

- Quickstart

- matsim-example-project

- matsim-code-examples

- MATSim versions and maven

Quickstart

`https://matsim.org/install`

We want to be able to run MATSimGUI and RunMatsim.

There are jogl (Java Open GL) problems :(... see URL for a fix:

`https://github.com/matsim-org/matsim-libs/issues/972#issuecomment-619370785`

Outline

2 Getting started

- Quickstart
- **matsim-example-project**
- matsim-code-examples
- MATSim versions and maven

matsim-example-project

The matsim-example-project pulls MATSim “from the cloud” (technically: from a maven repo).

→ 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.

[5-10 min]

Please help neighbors when done yourself.

Especially try if you can browse the MATSim source code even when not cloned/not open in IDE.

Outline

2 Getting started

- Quickstart
- matsim-example-project
- **matsim-code-examples**
- MATSim versions and maven

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/12.x`

[5-10 min]

Please help neighbors when done yourself.

Outline

2 Getting started

- Quickstart
- matsim-example-project
- matsim-code-examples
- **MATSim versions and maven**

MATSim versions, maven

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

```
<dependency>
  <groupId>org.matsim</groupId>
  <artifactId>matsim</artifactId>
  <version>11.0</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-example-project` for variants (snapshots, monthly snapshots, ...).

For experts: MATSim snapshot versions

If you want to switch to snapshots (unstable), the entry instead becomes:

```
<dependency>
  <groupId>org.matsim</groupId>
  <artifactId>matsim</artifactId>
  <version>12.0-SNAPSHOT</version>
</dependency>
```

This resides on a different maven repository, which thus also needs to be entered:

```
<repository>
  <id>ojo-snapshots</id>
  <url>http://oss.jfrog.org/libs-snapshot</url>
</repository>
```

For experts: MATSim snapshot versions, ctd.

Snapshots are built automatically (only) after all jenkins tests (see later) have passed.

They are refreshed on your local system typically once a day.

In consequence, a snapshot version may be somewhat older (in particular, when the jenkins tests do not pass).

We have found that programming against snapshot versions leads to difficult-to-debug problems; rather use weekly releases (see next).

For everybody: Weekly snapshot releases

If you want something in between unstable snapshots and oldish releases, there are automatic weekly snapshot releases:

```
<dependency>  
  <groupId>org.matsim</groupId>  
  <artifactId>matsim</artifactId>  
  <version>12.0-2019w40-SNAPSHOT</version>  
</dependency>
```

See [http://oss.jfrog.org/libs-snapshot!](http://oss.jfrog.org/libs-snapshot/)

Does not work if you need something even newer.

In that case, having matsim repo checked out into repo often better option than 12.0-SNAPSHOT.

See next ...

For experts: Dependency sharing in the IDE

If a maven dependency is available locally in the IDE ...

... and has the correct version (in the `pom.xml` near the top) ...

... then the IDE will link to that one instead of the version on the external repository.

May get quite confusing, since the local version can be inconsistent with the snapshot version “in the cloud”.

This is why we said “close all other projects in the IDE” above.

OUTLINE

- 1 Introduction
- 2 Getting started
- 3 MATSim scripts in Java**
- 4 Guice in MATSim
- 5 Regression tests
- 6 Conclusion (of this part)
- 7 Running on the math cluster and other remote compute engines
- 8 Plans innovation, plans removal, diversity
- 9 Train vs road evaluation in BVWP
- 10 Move on to java13
- 11 Emissions contrib
- 12 Visualization using VIA
- 13 Dashboards etc.: Towards automatic results presentation?
- 14 Visualization and dashboards using lavascript

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - Generate network in code
 - Dynamic factories
 - Time dependent network in code
 - Population generation in code
 - Guice ... later
 - MATSim GUI

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  
  
// ---  
  
Controller controller = new Controller( scenario ) ;  
  
// now modify controller if you want/need  
  
// ---  
  
controller.run() ;
```

Todos

Todo: Write an “empty” script.

Todo: Convince yourself that it runs iterations (!!!!).

Config options can be changed in code. Try

```
config.<tab completion>
```

as well as

```
config.controller().<tab completion>
```

Todo: Reduce number of iterations to 2.

[5-10 min]

Please help neighbors when done yourself.

Java as scripting language

- Java not optimal for scripting (wordy, messy to use from command line).
- However, did not want to introduce another language.
- May not hold forever. (TUB Maths has now switched to python; Thibaut Dubernet @ IVT has written a JPyte interface to MATSim.)

Sequential construction

- Computing, and by extension MATSim, have/had a lot of lazy initialization: Initialize only when needed.
- Too difficult for complex situations (like MATSim).
- In consequence, now try to build sequentially.
- Assume that config is immutable after inserting it into scenario.
- Scenario, however, not fully immutable: E.g., adding/removing plans ingrained into MATSim design. But some things are locked [[see Impl]].

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - Generate network in code
 - Dynamic factories
 - Time dependent network in code
 - Population generation in code
 - Guice ... later
 - MATSim GUI

Static factory methods

Something like

```
... ScenarioUtils.loadScenario( config ) ;
```

is a **static factory method**.

Factory methods are creational methods, they create objects.

Factory methods and factories (= classes which organize factory methods) will be discussed at several places.

Static factory methods vs. constructors

Static factory methods similar to constructors.

Reasons for using them:

- Can be more expressive in what exactly they do.
(Here: Also load the scenario from files.)
- Can return an interface rather than the implementation.
(Here: Return `Scenario` instead of `ScenarioImpl`.)
- Own memory management.

Reasons against using them:

- Potentially more difficult to find (not together with implementing class; no standard syntax).
In MATSim, try to compensate with `XxxUtils`.....

Look into Bloch, Effective Java.

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - **Code examples**
 - Generate network in code
 - Dynamic factories
 - Time dependent network in code
 - Population generation in code
 - Guice ... later
 - MATSim GUI

Stable code examples

Code examples by copy-and-paste are highly unstable.

Maybe ok in lectures/tutorials (if diligently reworked every year) ...

... but not ok anywhere else.

Way out: Provide runnable code examples that can be regression-tested.

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

Default branch is a version (i.e. not master). → stable 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.

How to find stable code examples

How to find code examples? Current variants:

- See in `matsim-code-examples` project.
Class that contains the code snippet often named `RunXxx` or `RunXxxExample`.
- Maybe have “approximate” code snippets in book or handouts; *always* point to code snippet inside repository for full version.

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - **Generate network in code**
 - Dynamic factories
 - Time dependent network in code
 - Population generation in code
 - Guice ... later
 - MATSim GUI

Generate network in code

Is there an example?

If not, let us write one *now*.

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - Generate network in code
 - **Dynamic factories**
 - Time dependent network in code
 - Population generation in code
 - Guice ... later
 - MATSim GUI

Dynamic factories

(normally probably just called “factories”; should maybe be called “polymorphic” factories)

```
Network net = scenario.getNetwork() ;  
NetworkFactory nf = net.getFactory() ;  
Node node = nf.createNode(...) ;  
Link link = nf.createLink(...) ;
```

`createNode(...)`, `createLink(...)` are dynamic/polymorphic factory methods.

The class containing these methods is the factory.

Note: `NetworkFactory` is an **interface**.

That is, the implementation behind it can be replaced.

Dynamic factories in Scenario

Specifically: Current code design would allow for

```
Scenario sc = OtherScenarioUtils.createScenario(...) ;  
//  
Network net = sc.getNetwork() ;  
NetworkFactory nf = net.getFactory() ;  
Node node = nf.createNode(...) ;  
net.addNode(node) ;
```

The links/nodes would be different implementations than above, behind the same interfaces.

Possible application: Different backings of the objects, e.g. by a database, with a JNI-bridge to C/C++, or in protocol buffers.

Dynamic factories in Scenario, ctd

Does *not* work any more if you program as follows:

```
Scenario sc = OtherScenarioUtils.createScenario(...) ;
Network net = sc.getNetwork() ;
// Node node = new NodeImpl(...) ; // no longer possible
Node node = NetworkUtils.createNode(...) ; // not recommended
//
net.addNode( node ) ;
```

node would be of wrong implementation type.

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - Generate network in code
 - Dynamic factories
 - **Time dependent network in code**
 - Population generation in code
 - Guice ... later
 - MATSim GUI

Time dependent network in code

Do we have an example?

How does it work?

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - Generate network in code
 - Dynamic factories
 - Time dependent network in code
 - **Population generation in code**
 - Guice ... later
 - MATSim GUI

Population generation in code

Do we have an example?

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - Generate network in code
 - Dynamic factories
 - Time dependent network in code
 - Population generation in code
 - **Guice ... later**
 - MATSim GUI

Outline

- 3 MATSim scripts in Java
 - Scripts-in-Java
 - Static factory methods
 - Code examples
 - Generate network in code
 - Dynamic factories
 - Time dependent network in code
 - Population generation in code
 - Guice ... later
 - **MATSim GUI**

MATSim GUI from MATSim example project

(already had this)

MATSim GUI part of example 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. `[[↔ mvn -Prelease?]]`

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

See <https://github.com/matsim-vsp/matsim-berlin> for an example.