

Getting started with MATSim

Technical introduction to the framework

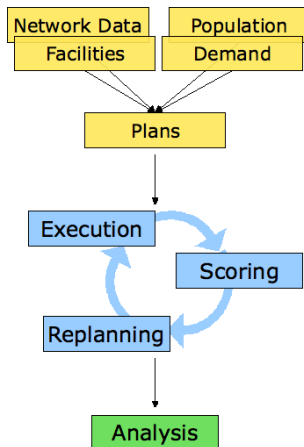
Dominik Grether

Transport System Planning and Transport Telematics
Berlin Institute of Technology

19.02.2008

Introduction

- MATSim toolbox for large-scale agent-based transport simulations
- Several modules
 - Demand modeling
 - Mobility-simulation
 - Re-planning
 - Controller for simulation runs
 - Analysis tools
- Modules can be replaced by own implementations



Introduction

What you'll learn

- How to run and simulate the provided sample scenario
- To understand the configuration settings so you can change them correspondingly for your own scenarios
- How to integrate an external, custom mobility simulation
- How to integrate a custom, external re-planning module
- How to analyse simulation results

Introduction

Requirements

- Java 5 or higher – Java Development Kit, JDK, Java SE (<http://java.sun.com/javase/downloads>)
- A subversion client (<http://subversion.tigris.org/>)
- An IDE, Eclipse recommended (<http://eclipse.org>)
- A subversion plugin – depends on IDE
- For Eclipse: Subclipse (<http://subclipse.tigris.org/>) or use Eclipse update site: http://subclipse.tigris.org/update_1.0.x

Introduction

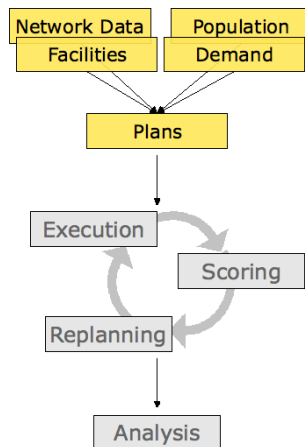
Checking out

- Command line:
 - `svn co https://matsim.svn.sourceforge.net/svnroot/matsim matsim`
- Eclipse:
 - File → New Project
 - Category SVN → Checkout Projects from SVN
 - URL:
`https://matsim.svn.sourceforge.net/svnroot/matsim`
 - Folder: `matsim/trunk`
 - Configure project using wizard
 - In wizard select “new java project”

Scenarios

Description of a Scenario

- Parts of Scenario:
 - Network: Road network
 - Population: Description of agents
- Configuration of scenario by XML-File



Scenarios

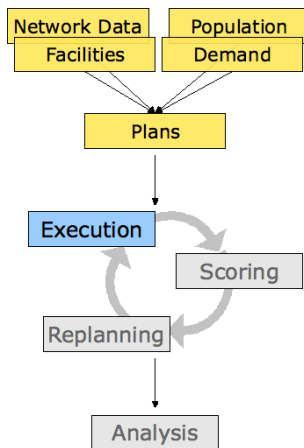
Equil Network

- Equil scenario:
 - `./examples/equil/`
- Start visualizer
 - Main method in `org.matsim.utils.vis.netvis.NetVis`
- Take a look at network
 - `./examples/equil/network.xml`

Running the examples

Running a single iteration

- 100 Agents from link 1 to 20
- Later from link 20 to 1
- Have a look at `equil_plans.xml`
- Run `org.matsim.run.Controler` with argument `examples/tutorial/singleIteration.xml`
- Examine the log for errors



Running the examples

Visualizing the simulation results

- Start Netvis again
- Open file `output/ITERS/it.0/SnapshotCONFIG.vis`
- Change daytime to 06:00 o'clock
- Increase linewidth
- Press play
- Read corresponding events at
`output/ITERS/it.0/0.events.txt`

Running the examples

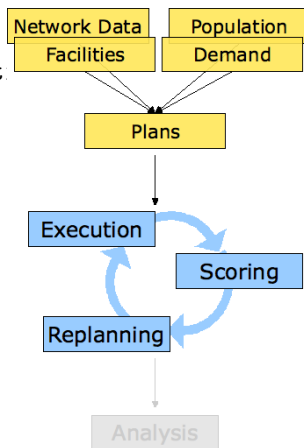
Modifying the settings

- Open `examples/tutorial/singleIteration.xml`
- Try to change settings in the module simulation, e.g. `endTime` of 07:00 or `snapshotperiod`
- Run the simulation again
- Be aware of the error:
The simulation will not overwrite files
- Make snapshots in “googleearth” mode

Running the examples

Running multiple iterations

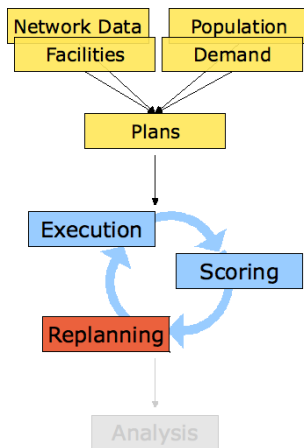
- Use configuration
examples/tutorial/multipleIterations
- Run controller
- 10 Iterations run with 10 % agents replanning
- Take a look at results
- Compare configuration files
- Increase number of iterations



Running the examples

Modifying the re-planning

- Change ModuleProbability_2 in multipleIterations.xml to 0.9
- Change ModuleProbability_1 to 0.1
- Run simulation again and look at results
- Replace value of Module_2 with TimeAllocationMutator
- Examine results
- Combine the 3 re-routing strategies



Creating a custom controller

Introduction

- Custom controller for
 - Integration of own code
 - Customized analysis
 - More complex scenarios which require special modules
- Only rewrite the parts you need, but try!

Creating a custom controller

First, helpful steps

- Inherit from `Controller` and add `main(String[] args)`
- Create an instance and call `setOverwriteFiles(true)`
- Call the `Controller.run()` method
- Open visualizer after `run()` is terminated

```
String[] visargs = {"../output/ITERS/it.0/Snapshot"};  
NetVis.main(visargs);
```

Creating a custom controller

Handling simulation events

- Simulation events output of simulation (physical world)
- Controller attribute protected final Events events
- Handler interfaces in `org.matsim.events.handler`

Creating a custom controller

Handling simulation events

- Write own handler

```
MyHandler implements EventHandlerLinkLeave {  
    public void handleEvent (EventLinkLeave event) {  
        ...do something...  
    }  
    public void reset(int iteration) {  
        ...reset something...  
    }  
}
```


Creating a custom controller

Handling simulation events

- Create an instance

```
MyHandler handler = new MyHandler();
```

- Register at Events instance

```
events.addHandler(handler);
```

Creating a custom controller

Handling controller events

- Controller events output of simulation process, e.g.
 - Startup complete
 - Begin iteration
 - End iteration
 - Replanning
 - Shutdown

Creating a custom controller

Handling controller events

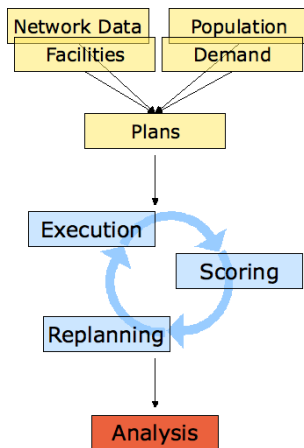
- Interfaces in `org.matsim.controller.listener`
- Implement and add by

```
Controller.addControllerListener(myListenerInstance);
```

Creating a custom controller

Analyse results

- Controller generates some analysis
 - Score statistics
 - Plans (per 10th iteration)
 - Snapshots (per 10th iteration)
 - Leg histograms (per iteration)
- Customized charts via Event handler / listener
- Useful tool:
`org.matsim.utils.charts`



Outlook

Next steps for Advest

- VSP will provide 1%, 10%, 100% input data for
 - Traffic light control
 - Evacuation scenarios
- VSP runs big scenarios
- Further questions and tasks?