

User Manual for Transportation Mode Module 1.0

Jianqiu Xu

April 23, 2012

To run the program in SECONDO, three setups have to be completed.

1. SECONDO Environment

- Install the database system SECONDO. All information refer to the web:

<http://dna.fernuni-hagen.de/Secondo.html/index.html>

- Please use the secondo source package on the TransportationMode web page (given below) instead of the one from SECONDO home page. The source package includes the standard SECONDO configuration and all algebras needed by Transportation Mode Module.

Transportation Mode Web Page: <http://dna.fernuni-hagen.de/secondo/TransportationMode/TM.html>

- The following algebras have to be activated: TransportationMode, Network, TemporalNet, Traffic, PlugJoin, SpatialJoin. Make sure that the following lines are added in makefile.algebras (secondo/bin)

```
ALGEBRA_DIRS += TransportationMode
```

```
ALGEBRAS += TransportationModeAlgebra
```

```
ALGEBRA_DIRS += Network
```

```
ALGEBRAS += NetworkAlgebra
```

```
ALGEBRA_DIRS += TemporalNet
```

```
ALGEBRAS += TemporalNetAlgebra
```

```
ALGEBRA_DIRS += Traffic
```

```
ALGEBRAS += TrafficAlgebra
```

```
ALGEBRA_DIRS += PlugJoin
```

```
ALGEBRAS += PlugJoinAlgebra
```

```
ALGEBRA_DIRS += SpatialJoin
```

```
ALGEBRAS += SpatialJoinAlgebra
```

- Turn off transaction. Please activate the line “RTFlags += SMI:NoTransaction” in the file SecondoConfig.ini (located in the directory: secondo/bin).

2. Input Data

- Raw Data: (1) Berlin Roads; (2) Houston Roads; (3) public floor plans. (The names for roads have to be “berlinroads” and “houstonroads”. Please do not change them, also for floor plans.)

- Auxiliary data: (1) a relation for streets speed; (2) indoor paths; (3) traffic relations (these data are created by the program and will be used in the running)
- Steps to prepare the data.
 - Download all the above data from the website.
 - Create a directory named **TM-Data** in `secondo/bin` and put all data into that directory. Do not forget to unpack the packages for floor plans and auxiliary data. Put all files of auxiliary data in the directory `TM-Data`. For floor plans, put the file folder (`FloorPlan`) in the directory `TM-Data`.

3. Running Scripts

- Download the script package from the web page and put all files in the directory `secondo/bin` after unpacking.
- Steps to execute the scripts:
 - Create infrastructure data. There are two files: (1) `createberlingenmo.sec` and (2) `createhoustongenmo.sec`. The first one creates the data for Berlin and the second creates the data for Houston. After starting `SECONDO` terminal, type `@createberlingenmo.sec` to create Berlin and type `@createhoustongenmo.sec` to create Houston. More files are used to create the infrastructure data: `createpave1.sec`, `createpave2.sec`, `createbus.sec`, `createmetro.sec` and `createindoor.sec`. These files are automatically called by `createberlingenmo.sec` and `createhoustongenmo.sec`. Users do not have to explicitly execute them.
 - Create benchmark data. After creating the infrastructure data, the system is able to generate moving objects. Start `SECONDO` terminal, type `@createberlinbench.sec` to create moving objects in Berlin and type `@createhoustonbench.sec` to create moving objects in Houston. In the first few lines, there is a variable called `bench_scale` which is used to set the data size. One can change the value to create the required data. (By default, the scale factor is one and the program will create 10,000 trips.)
 - For each benchmark data, there are three files to run the queries.
 - Berlin: (1) `berlinbenchIF.sec` (2) `berlinbenchMO.sec` (3) `berlinbenchMO2.sec`.
 - Houston: (1) `houstonbenchIF.sec` (2) `houstonbenchMO.sec` (3) `houstonbenchMO2.sec`
 The first file is for infrastructure queries and the second and third are for moving objects queries. The second file does not employ the optimal techniques and the third does. After executing all queries, one can type “`query bench_res1`” and “`query bench_res2`” to see the query cost for each query. Two relations are created in the database to store the query results, `bench_res1` and `bench_res2`.
 - The file `benchbenchMO3.sec` runs all queries in the data model report using the data Berlin.
- 3D viewer is only used to visualize indoor objects, but it is not required to create the data and run the queries. Get the description file on the web page.

Environments

- AMD 3.0 GHz, 4 GB memory, 2T Disk, Suse Linux 11.3 (32 bit), G++ 4.5.0

All experiments and scripts have been tested in this environment, including generating the infrastructure data, producing different sizes of moving objects and running benchmark experiments.
- AMD Phenom II X6 1055T, 8 GB memory, 1T Disk, Ubuntu 10.04.2-LTS-Desktop (64 bit), G++ 4.4.3

The procedure of generating the infrastructure data is tested. Since the tool is able to create different sizes of moving objects, the program is tested on some scale factors. Benchmark queries are already tested.

P.S. In the second environment, an error might show when compile the secondo source code, “cannot find -loptparser”. If this happens, goes to the directory secondo/OptParser, type (1) “make clean” and (2) “make”. Then, compile the code again and it should work.