Metro Network Journey Simulator v3

Key contact Stephane Duguet, Thales GTS HK

Objective:

Incidents occur from time to time in the railway network, e.g. failing equipment, track obstruction, passenger incidents, which cause train delay and severely disrupt the service.

Thales already provide data analysis on passenger flow, but we would like a way to generate simulated but realistic data from parameters to

- simulate an imaginary transport network, or an existing network where we don't have the data
- generate incidents data that are seldom present in real datasets

In 2018-2019 and 2020-2021, ESEO students developed a first version of the Metro Network Journey Simulator.
We are interested to have a new team from ESEO improve that solution to improve the solution with for example less flat randomized result and more parameters to generate trips.

Technology

The technology used is

- Golang for the simulator
- Java for the railway editor HMI.

Note : The core of the work will be done in Go.

Tasks

Understand previous work.

List of features that have to be developed:

Simulator (priority 1):

1. Make the simulator more realistic:
   - Generate less uniform random data
     - See two attached picture at the end of the document of analysis done on resulting files using QlikSense for reference, that have too clean patterns from random values
     - See sample time series patterns at the end of the document from real data
   - More trips types, or population type (Currently only commuters / non-commuters that impact Peak / Non Peak hour), perhaps have the railway-editor help to setup % of different zones: Residential, Commercial, Office/ Industrial, Touristic (to be defined) … per station. And manage trips according to that percentage and the volume of population type.
     The common Origin & Destination (O&D) pairs for commuting traffic include:
     - Residential >> Commercial
     - Residential >> Office
     - Residential >> Industrial
     - Commercial >> Touristic

     Invent (brainstorm and come up with) assumptions, rules and required parameters, ex :

    - For a population of 500,000 with +5% tourist, Tourist will sleep in Commercial area, the rest in Residential area.
    - People in residential area will have to be split in the station that have a residential %, balanced by the % of zone.
    - The part that commute will do so to working zone (Commercial, Office, Industrial)

2. Manage a proper calendar:
    - Manage different type of days : Weekday / Weekend / Holiday
        - Perhaps add smaller peak (lunch during weekday, evening nigh peak the Friday ...?)
        - Add different pattern for week-end
    - Add special events to the calendar (date / time period / impacted stations / pattern change), for example sport event, concert, or protest. Clarify if that is a different or same handling than Incident events?

3. Modify the trains schedule according to the time of the day
    - Have train reset during the night closing, and be sure all passenger exit before closing (no passenger also enter that can't exit)
    - Launch more train to cover peak hour,
    - Check train distance and proper spacing on the line,

4. Check that the incidents works as expected
    - for all events manage properly what happen to the population
    - Clarify if that is a different or same handling than Incident events?

5. Improve memory management and performance for computing many days (reset each night?)
    - Being able to run multiple day simulation
    - cutoff at business day end

6. Containerize the simulator and possibly the application (linked to have a web version of the app)
    - Using dockerfile and docker-compose

    It's nice to have feature

<u>Railway-Editor (HMI):</u>

1. test maximums (stations number (> 500) lines number (>40))
2. Improved HMI for creation / suppression / of station and lines
3. Improved HMI for event management
4. Configure Zone percentage: Residential, Commercial, Office/Industrial, Touristic with check of total? Perhaps create zones of one type with names, and in another screen balance/set the %?
5. Have all the configuration manageable from the HMI
6. Running the simulator from the HMI, with feedback and result integrated too
7. Add configuration versioning / result and data produce versioning.
8. Web version (nice to have)

<u>Dashboard</u>

Redo and improve the dashboard for aggregate test using PowerBI instead of QlikSense as there is no more unlimited free version for this last one.

For reference

Example of timelines for a day of data produced by the simulator V1 :

Example of timelines for a day, for 1 entry station, of data produced by the simulator V1 :

Example of timelines for a day for different entry stations of real data: