

Some notes on the railwayYung1995 model - 28/10/05

This model illustrates a number of aspects of EM development.

1. Animation from LSD using the ADM

One component of the model referred to in the README is the train arrival and departure model to be found in

~wmb/public/projects/simulations/train

This is based on the animation of the LSD specification that has already been introduced in the lectures.

The train model illustrates how an LSD specification may be hand-translated into the ADM, then translated into tkeden.

There are some auxiliary notes in

~wmb/public/projects/simulations/train/noteswmb.txt

- these give details of how the observables associated with agents in the LSD specification are represented in tkeden. For instance, pat_1 is the station at which passenger 1 is currently, and pos_1 indicates their position on the platform.

startClock=1; is the tkeden trigger to start the ADM animation.

2. The railway track model

The train model is run in conjunction with a Donald/ARCA model of a railway track. Note that you have to enlarge the window to get access to the control buttons. Manual operation of trains is possible without starting the ADM animation linked to the train model. Train 1 is semi-automatically controlled - it is linked to the train model, so it is simpler to exercise Train 2 manually - this is the train initially on the inner track.

3. Negotiating conceptual integrity

There is ambiguity here about whether the model represents a model railway, a model of the track segments displayed in a real signal box, or is intended for studying train arrival-departure protocols. One virtue of the model is that it allows all these possible viewpoints to be exercised concurrently, with a view to eventually establishing a greater degree of conceptual integrity. The development of model to this point illustrates one or two interesting example of observational issues that arise when extending and combining models.

Appendix - noteswmb.txt from ~wmb/public/projects source.

The model can be run in two ways: by including train.e or train.s

Under both executions, should use startClock=1; to start the animation.

By invoking train.e simply get a commentary on the action, whereas train.s leads to a visualisation.

pat_1 is location of passenger 1

current value of at is location of train

boarding_1 indicates that passenger 1 is boarding
alighting_1 that passenger 1 is alighting

can_move is the objectively defined observable indicating wheher the train can move

pos_3 == 1 --> on platform

pos_3 == -2 --> seated in carriage

This demo was developed by Simon Yung, and combines the animation of the train arrival and departure protocol that is to be found in \$PUBLIC/projects/simulations/train with a simulation of a model railway layout. The model of the track interconnection was constructed using ARCA, but the tkeden variant makes use only of the translated ARCA code. (The original ARCA code is in the projects/railway directory.)

To execute the animation, it is necessary to open up the Scout screen window to reveal train control buttons (below the track layout), and the visualisation of the stations (to the right of the track layout). The arrival and departure protocol is implemented using the ADM, so needs to be invoked using

```
startClock = 1;
```

- the visualisation depicts the protocol as it operates for a train leaving the station at the top of the picture destined for the station at the bottom of the picture.

The locations of the three stations (A5, A10, B5) are not explicitly indicated on the track layout. The references are to the donald names of track pieces: the outer track comprises pieces A1-A10, and the inner track B1-B10.

The train that is depicted in the arrival/departure protocol model corresponds to Train 1 in the animation. Note that, as a result, Train 1 is in part operated automatically: it will in general leave a station as authorised by the stationmaster in the same direction as it entered it. In order to get the train to travel between certain scheduled stations (e.g. to get from A10 to B5), it is necessary to control the train using the model railway buttons. (There are sometimes problems of interference in this transfer of control between agents.)

The auxiliary files rail.supp and tracktest.d give more insight into the model. The redefinitions in rail.supp illustrate substitution of 1 track piece for another; the mechanism used to change points in ARCA; expanding the width of the track; the suppression of the ARCA diagram; the control of trains without the button interface; train relocation; and changing the mind of the passenger and driver re destination. The file tracktest.d is a simple illustration of an experimental environment extracted from the model - it helps to indicate how the track model is constructed (and could be changed). The potential and limitations of donald as an object modeller are also revealed in this extract.

The speed of the simulation can be reduced by setting the variable Pause to a positive integer value (e.g. Pause=2).