A Large Mass Transit System With Failures

0207147

Abstract

In the field of Empirical Modelling, much has been written and many models created relating to transport systems. However, most of this work has focussed on small-scale systems of a few manually modelled places. There is much to be gained from taking a wider view of a transport system and how it reacts to local conditions. The model accompanying this paper attempts to do so, including various kinds of failures, for a particularly topical example: the London Underground, where there have been an increasing number of security alerts in the wake of the 7th July attacks.

1 Introduction

In this project, the aim was to model the London Underground network (hereafter referred to as LU) in EDEN[1], and thereby investigate the feasibility of, and best approaches for, this kind of model in this environment. Its potential uses outside that investigation include demonstration of various properties of transport networks like the LU, and, for modellers, modelling the likely effects of potential modifications to the LU.

The particular focus of this model is investigation of the effects of four kinds of localised failures upon the network as a whole. As such, the locations of stations, the routes (Lines) that connect them, and the behaviour of trains have been modelled, but many other details, such as train capacity, signalling, depots and servicing of trains and track have not. As is explored in more detail in section 3, the LU is a complex system and the capabilities of EDEN on a desktop PC are finite.

The model has aimed for realism wherever real data is available and performance permits, in order to simulate as accurately as is practical the likely effects of the conditions being investigated on the real-world LU.

2 Model Development

2.1 Design Decisions and Features

The main property of the network displayed in the model is delay, and as such, it was important to begin development by laying down a network with realistic distances between stations, in order for travel times also to be realistic.

Fortunately, a machine-readable LU map[2] was available (under the GNU Free Documentation License) on Wikimedia Commons. This is based on measurements made outside each station using the Global Positioning System, and includes not just station positions but the structure of all the lines.

Another crucial decision taken at the start of development was which perspectives should be available to the user of the model. In order to show the state of the system in a way that is quick and easy to interpret, an interface broadly similar to that of air traffic control was chosen. This consists of a map of the LU stations at their correct geographic positions (unlike in the official map), with lines between them in the familiar colours, and identification numbers at the positions of the trains. This shows at a glance the relative distribution and speeds of trains, including phenomena such as the lower average speed in areas with many stations, and the queuing that occurs behind a slow train.

These are described as relative speeds because the clock in the model runs at approximately 10 times real time. This is done in order to see the onset of effects without waiting minutes or hours, and to allow the system to reach full operation on start-up (which is the start of the day) reasonably promptly.

The speeds are lower in areas with more stations because the trains are modelled as having relatively low acceleration, and thus do not reach top speed in these areas. While this is not strictly accurate (speeds are actually limited on tighter curves and more complicated points for safety), it does give approximately the correct speeds across the network.

The factors that stop a train from proceeding at full speed are: nearing the next stop; trains ahead; a train on the same route at the next stop travelling in

---

[1] E. Yung, 1989

[2] Sanders and Forrester, 2005

a sufficiently similar direction to be considered to be occupying the same platform (causing the later train to wait outside the station).

The clock in the model runs for 24 hours, and the timetable models a single Thursday, looping forever, although the real-world timetable does not vary much between weekdays[3]. The timetable is based on Transport for London's published first and last train times[4], modified and added to in order to approach Tubeprune's train counts[5], which, published amongst a range of other statistics collected by a former LU driver, give a fair indication of the number of trains that should be present on any given line.

For added realism, a satellite image of London is provided underneath the map, lining up with the positions of the stations[6], but faded slightly to avoid it being distracting. This shows landmarks and, to an extent, population density, and hence which stations are likely to be more heavily used than others.

As well as panning around the map and watching the system, the user can select trains and stations to display information about them, and, taking on the role of a passenger, make a journey. This can be done by selecting a station and using the buttons provided to board a train that is stopped at the station, subsequently leave that train at a different station, and maybe then board others. When doing this, the user will notice that the information displayed for a train is as close as possible to that actually announced on board real LU trains. The user is of course more capable than a real passenger in this role, since they can select a different station or even a train in motion at any time.

Finally, taking on the most powerful role, the user can cause trains to slow or stop, and stations to close. These options are intended to represent several likely failure modes of the LU. A minor motor fault, or in overground areas, leaves on the line, might cause a train to slow down. A total motor failure, or problem on the line ahead, would cause a train to stop.

The two modes of station closure are “no stopping” and “closed”, with the latter preventing trains from advancing beyond their previous station. These simulate a station that is closed for building works (no passenger access, but trains may pass through) and a station with a serious fault or hazard such as a security alert, respectively.

2.2 Implementation Details

This section will cover the methods by which this system, differing as it does from most previous EDEN models, was implemented in EDEN.

The main challenge of this model was that all objects present on the map (other than the background) are members of large collections of similar objects (stations, trains, edges). Such collections are traditionally represented programmatically using arrays, which are not available in the line-drawing language of the EDEN environment, DoNaLD[7].

The approach used here to overcome this is EDEN's `execute` function, which allows strings to be executed in any language. For each of the stations, whose positions are fixed when starting the model, a DoNaLD point is created corresponding to the station's identity number. Definitions of tracks then depend only on the station points, and similarly the trains, defined as labels, are redefined each time they pass a station, and express their position as proportionally between the relevant two stations, the proportion being an EDEN variable linked by dependency to how much of that distance they have travelled.

It is tempting to suggest the use of DoNaLD's graph feature as a more elegant alternative, but it seems able only to produce many objects for many calculations of a DoNaLD formula with different input values. As such, it still lacks the ability to index into an EDEN array. A helper function could have been used, but this would have been slower than the current approach and arguably no more elegant or readable.

The performance graph in the lower right of the interface is implemented by tracking what proportion of trains, during each clock tick, were applying their brakes (“blocked”). A level of around 10% is expected due to trains approaching stations. The graph itself is again drawn using DoNaLD, but with a definition for each data point in the history period that extracts a value from the array in order for DoNaLD to be able to access it. This has the advantage that the graph is changed simply by changing the values in the array.

The bar showing the proportion of trains released and running on the network is implemented similarly, but controls the width of a SCOUT window. This meter is present mainly in order to remind the user that the time of day will affect how many trains are expected to be present.

A final note on station data: should the LU be altered and a new version of the LU geographic map data released (notably, the DLR extension for London City Airport is already open, although DLR

---

[3] Thursdays were chosen because they offer the excuse that the author never could get the hang of them (apologies to Douglas Adams).

[4] Transport for London, 2005

[5] Tubeprune, 2005

[6] This is most easily verified at Heathrow (at the south-west end of the Piccadilly line), and at the stations along the edges of the river and Hyde Park.

[7] Beynon, Angier, Bissell and Hunt, 1986

trains do not currently run in this model), the CSV (Comma-Separated Values) files in the data directory can be replaced with new ones, as the stations are loaded on each run from these files, which are exactly as originally presented on Wikimedia Commons.

3 Difficulties and Changes

3.1 Language Limitations

The movement of trains, including calculations for variable velocity, the redefinition of their labels, and related tasks, is handled by a function called `moveTrains`, which is unfortunately somewhat large. This was difficult to avoid, given the interconnectedness of the various decisions affecting trains, and thus the number of values that would need passing between smaller functions. However, care has been taken to ensure that this and all other functions are well commented, so that the model can be modified within the file.

It is recognised that needing to do this in order to make many interesting kinds of changes is suboptimal from the point of view of the Empirical Modelling experience provided by this model, and indeed that much of this code is perhaps overly procedural for the environment (as in not well suited to interactive modification in tkeden), but it is a result of an underlying problem with EDEN: the absence of object-oriented features. This means there are very few effective ways to model types (or classes) of objects, and procedural loops over arrays were felt to be the best of these choices for this application.

3.2 Performance

The original intention was for the model to include passengers, each travelling between two randomly-selected stations. It was suspected even at the time that modelling a realistic number of passengers (about 3 million per day[8], perhaps as many as 1 million active simultaneously) was beyond the reach of EDEN on a current desktop PC.

Unfortunately, the performance of the model even without them approaches the limits of usability, and it is for this reason that they have not been included. However, it has been possible to model almost the full LU network, rather than just Zone 1 as was originally planned, as this is less of an increase in requirements by several orders of magnitude. Regrettably, the Central, District, Metropolitan and Northern Lines have still had to be excluded.

Passengers are not completely absent, because, as noted earlier, the observer can act as a passenger, and thereby experience any typical journey that they select. This is, of course, a compromise.

4 Future work

4.1 On This and Related Models

Although care has been taken to produce as realistic a model of the LU as possible, with greater time and resources, a number of areas are open for further investigation.

The most obvious of these is passengers. On some subset of the network, they may be practical to model, although it should be noted that passengers do very often use multiple lines to complete a journey. With the modelling of passengers, a more accurate estimate of disruption to service can be obtained than by measuring disruption to trains. As the model stands, a blockage of a line will, once cleared, soon no longer affect the proportion of trains blocked. However, this does not take into account the inconvenience to the passengers of the trains that spent that time waiting.

If passengers are modelled, and their inconvenience can be quantified, it may also be of interest to predict the likely inconvenience (perhaps even in monetary terms) of an alteration to the LU or of building works required for an extension. A suitable model would allow such possible scenarios to be compared.

In the area of train behaviour, some approximations are made, and the relevant aspects could be investigated in more detail by dedicated models. Indeed, a possible departure protocol for trains already forms the basis of a prominent example model[9]. The LU departure protocol differs from this, and could be incorporated into this or other models, including, at its simplest, variable dwell times at stations depending on the numbers of passengers boarding and alighting.

Signalling is not modelled here, only collision avoidance on the basis of stopping distances. The real LU signalling system operates on the basis of blocks of varying lengths, as with main-line trains. There are systems and signals for detecting when one train has cleared a section and not allowing another to enter until this has occurred. These do not appear to have been modelled in EDEN.

In addition to the latter, the nature of the start-up and shut-down phases of the network, along with the differences in speed between different parts of the LU due to artificial low speed limits could also be investigated.

---

[8] Transport for London, 2003

[9] S. Yung, 1995

Finally, it would perhaps be interesting to investigate whether dtkeden[10] could be used to allow several machines to share the load of a more complex version of this model, without compromising its responsiveness or reliability.

4.2 On EDEN

As is hopefully obvious, this model's code would greatly benefit from the addition of stable, reliable object-oriented features either to EDEN or as an extension to it. That is this paper's recommendation if more models of a similar real-world referents to this one are desired, namely those of large systems comprising many similar objects.

When modelling a large system, if many objects are to follow the same rules, it is more natural to express those rules more independently of the current set of objects than is the case here. This would allow projects such as this one to be written more elegantly, readably, extensibly and easily.

Such features, if implemented, should ideally integrate well with DoNaLD and similar additional languages, or much of the benefit of adding these features will be lost due to the difficulty in communicating complex EDEN data structures to DoNaLD.

Acknowledgements

The author is indebted to Ed Sanders and James D. Forrester, without whose geographic data[11] this model (at least with its current scope) would have been much harder to create, as no other LU mapping projects[12] have published any machine-readable sets of co-ordinates.

The assistance of the author's friends and housemates, partly through occasional feature suggestions (but mostly in putting up with the author's frequent fretting) should also be acknowledged.

References

Edward Yung. EDEN: An Engine for Definitive Notations. MSc thesis, Department of Computer Science, University of Warwick. http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/mscbyresearch/eyung/, 1989.

W.M. Beynon, D. Angier, T. Bissell and S. Hunt. DoNaLD: A Line-drawing System Based on Definitive Principles. http://www.dcs.warwick.ac.uk/reports/086.html, 1986.

Ed Sanders and James D. Forrester. London Underground geographic maps. http://commons.wikimedia.org/wiki/London_Underground_geographic_maps, 2005.

Transport for London. History – Introduction. http://www.tfl.gov.uk/tube/company/history/, 2003.

Transport for London. First and last trains. http://www.tfl.gov.uk/tube/travelinfo/firstandlast/?line=all, 2005.

Tubeprune. London Underground Statistics. http://www.trainweb.org/tubeprune/Statistics.htm, 2005.

Simon Yung. Train and Railway. http://empublic.dcs.warwick.ac.uk/projects/railwayYung1995/, 1995.

Pi-Hwa Sun. Distributed Empirical Modelling and its Application to Software System Development. PhD thesis, Department of Computer Science, University of Warwick. http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/phd/psun/, 1999

Owen Massey. Mapper's delight: the London Underground Diagrams. http://owen.massey.net/tubemaps.html, 2005.

Tom Carden. Travel Time Tube Map (Built with Processing). http://www.tom-carden.co.uk/p5/tube_map_travel_times/applet/, 2005.

James Bow. The London Tube Map Archive. http://www.clarksbury.com/cdl/maps.html, 2002.

---

[10] Sun, 1999

[11] Sanders and Forrester, 2005

[12] Owen Massey, 2005; Tom Carden, 2005