

Mixing and matching!

Exploiting software possibilities in our processing and presentation of modelling data

SIMON DOYLE

Vela VKE

PO Box 72927, Lynnwood Ridge, 0040
tel: (012) 481 3842 fax: (012) 803 7943
email: doyles@velavke.co.za

with acknowledgements to:

David Turner and Tony Sharpe of Halcrow,
Dan Jones, an independent transportation modeller,
Serco,
and the Highways Agency

SUMMARY

Transportation modellers enjoy a growing range of software platforms with which to process and present the data they work with. Standard office packages like Excel and Access as well as powerful GIS packages like MapInfo and ArcInfo/ArcView are becoming as much a part of the working environment of transportation modellers as the modelling packages they are traditionally familiar with.

Although effectively independent of each other, and more-often-than-not the products of wholly independent institutions, most state-of-the-art software packages respect relatively simple and transparent data transfer protocols which allow for the easy transfer of data. In certain instances, formal interfaces exist that are either commercially available or custom written for a specific project or task. Simple table and graphical entity transfers, however, often allow data transfer without any formal interface. Thus, with or without formal interfaces, data transfer between independent software packages has become something that can be assumed rather than agonized over as too inconvenient or onerous to attempt. Accordingly, it's not so much how data can be transferred between packages that presents the challenge as what can be done when such transfer is assumed.

Apart from briefly emphasizing the point that data transfer is not the issue it's often made out to be, an unusual illustration of how a couple of standard off-the-shelf packages and one custom-written program were used in an integrated manner and to useful effect, with and without formal interfaces, will be presented. It comprised the route set generation exercise undertaken as part of the development of the United Kingdom's National Traffic Control Centre. In this exercise Emme/2, a stand alone and custom-written computer program and MapInfo were used in sequence to build a diversion route library.

1. INTRODUCTION

As computing powers and options increase, transportation modellers enjoy a growing range of software platforms with which to process and present the data they work with. Standard office packages like Excel and Access as well as powerful GIS packages like MapInfo and ArcInfo/ArcView are becoming as much a part of the working environment of transportation modellers as the modelling packages they are traditionally familiar with. In fact, a failure to exploit the broad range of options that exist, to some degree at least, can become a liability.

Unfortunately, the transfer of data between different software platforms is often imagined to be too difficult to be worth the effort unless some form of formal interface or formal software integration exists. The fact of the matter is that most state-of-the-art software packages respect relatively simple and transparent data transfer protocols which allow table and graphical entity transfers without any formal interface.

Apart from briefly emphasizing the point that data transfer is not the issue it's often made out to be, an unusual illustration of how a couple of standard off-the-shelf packages and one custom-written program were used in an integrated manner and to useful effect, with and without formal interfaces, will be presented.

2. DATA TRANSFER REALITIES

Appreciating the possibilities

If one of the biggest errors in modelling comprises human error, perhaps one of the biggest limits comprises a lack of human creativity. This affects all aspects of modelling. In this particular instance, however, a failure to exploit software possibilities, both in terms of modelling packages themselves but also other packages, is in view.

It must be granted that the precious commodities of time and budget are not always on the transport planner's side (although there is an uncanny way in which limited time and budget can actually demand "thinking outside the box"). Further, not every planning situation demands or even deserves imagination. Nevertheless, and purely as an observation, a lack of creativity in the use of available software plagues many a modelling exercise. Extended processing and presentation options have now become so accessible to the average planner – even the planner without specialist programming skills - that there is almost no excuse for not entertaining a very broad-minded approach to software use.

Perhaps one of the biggest obstacles to a broad-minded approach to software use is an outdated understanding of what is actually necessary to make data transfer a reality. Gone are the days where data transfer between separate and even diverse software packages is the exclusive province of the specialist programmer. Although effectively independent of each other, and more-often-than-not the products of wholly independent institutions, most of the state-of-the-art software packages seen in the average planning office or accessible to the average planner respect relatively simple and transparent data transfer protocols which allow for the easy transfer of data. In certain instances, formal interfaces exist that are either commercially available or custom written for a specific project or task - the Aimsun2/Emme/2 interface being a case in point. Simple table and graphical entity transfers, however, often allow data transfer without any formal interface. Thus, with or without a formal interface, data transfer between independent software packages has become something that can be assumed rather than agonized over as too inconvenient or onerous to attempt. Accordingly, it's not so much how data can be transferred between packages that presents the challenge as what can be done when such transfer is assumed. Of importance is a basic acquaintance (c.f. mastery, which is not a necessity, especially if one works a large planning circle) with what is possible and a boldness to explore and experiment. Obviously, this is easier said than done, because (for most of us, at least) it's too tempting to stick to tools we know and are skilled in rather than seeking out and exploring the best tool for the job. The best tool for the job might very well comprise an unknown function in the package we are already using!

Motivation

There are at least three reasons why a broad-minded approach to software use should be cultivated by transport planners:

- First, current design and planning environments encourage it – in the sense of both the origin and the destination of design and planning datasets.
- Second, the expectations and capacities of the public and political and technical – level decision-makers demand it.
- Third, planning exercises invariably arise where the powers of a mix of independent but complementary packages or programming platforms are desired if not demanded.

Data sources and destinations

In the first instance, transport planners receive and use data originating from or formatted according to a fairly wide range of software sources. There is a need to not only access this material, but also transform it into the formats required. More often than not, the exercise of effecting the transformation is straightforward. Standard graphical entity and attribute formats are fully-interchangeable. Further, the tabular structure of GIS databases or spreadsheet datasets all but guarantee easy table transfer. In fact, data transfer in the transport planning environment involves little more than table or graphical entity transfers. Every once in a while, however, obstinate datasets cross the tables of planners which demand a little bit more initiative.

note:

The presenter recalls receiving three years' worth of accident reports for a fairly large and populated area in a heavy box of computer printout (i.e. hardcopy format). The initial shock of receiving such a large amount of pure ink on paper paled into insignificance compared to the supplier's insistence that the vintage software it was generated by could produce nothing else without specialized software alterations. Brief discussions with the IT department of the institution involved revealed that there was at least one option available through which an electronic format of the accident reports could be generated. This comprised getting the vintage software to print to (ascii) file rather than print to printer. Accordingly, an extremely large electronic file arrived on CD. Relieved, but not totally out of the woods, the heavy box of printed material was put aside and thoughts engaged on how to transform the ascii material into a more useful and interrogatable format.

Two options presented themselves – first, write a little bit of code to “parse” the lengthy ascii file, identify selected data types and then write them to a database – second, use Excel's marvellous array of functions (IF, VLOOKUP, LEFT, RIGHT, INT, etc.) to effectively do the same. Time and capability constraints (namely, the presenter's limited programming skills) argued in favour of the latter. Within hours a wholly electronic but sifted and structured version of the ascii file material existed. A day later, the integrity of the captured data had been adequately confirmed by a number of simple validation checks. Pivot table manipulations did the rest of the work.

Demanding audiences

In the second instance, transport planners find themselves increasingly exposed to parties whose appreciation of planning technicalities and subtleties demands “the second mile” in terms of presentation and explanation effort. The ability to quickly move data between software platforms and generate alternative and/or supplementary visual material which is both comprehensible and persuasive becomes extremely useful. It's not that the traditional formats are fatally-flawed or are becoming redundant in any way – on a working level, the traditional assignment plot will probably always comprise the first stop for the planning team. Nevertheless, the graphical possibilities afforded by Emme/2's Enif, GIS platforms like MapInfo and ArcInfo/ArcView or even freely-available GIS packages like AccuGlobe become helpful allies in the communication game.

It goes without saying, that the extended visualization or processing powers of available software packages does not in itself guarantee the generation of either useful or comprehensible material. Generally speaking, dedicated effort and thought are required to generate and configure good presentation material – the wisdom of the best material often lying in what's left out. In fact, the time needed to generate useful graphical material lies in the “what” rather than creating the mere possibility through table and shape file transfers.

note:

The presenter has found it best to continually explore possibilities as a project progresses and as data availability allows rather than in a last ditch sprint at the end. The ridiculous time scales many of us work to (in South Africa, at least) must have an effect on what we manage to do in terms of creative and effective data presentation. Time is a necessary ingredient.

Unusual data processing needs

In the third instance, while integrated visual presentation tools (part-and-parcel of the basic software or independent but parallel entities to it) go a long way, if not all the way, to satisfying needs, some type of extended, versatile and flexible software use invariably becomes needed, especially in large or unusual modelling studies. The simple geo-referencing powers of standard GIS packages are a case in point (i.e. to automatically determine aggregation relations).

Since the example addresses the subject of unusual data processing needs, no more will be said about this issue at this point.

Basic rules

In all of the above instances, a few well-worn but also proven principles deserve emphasis, namely:

- do no more than is necessary,
- develop simple one-stop correlation tables,
- build in regular and diligent book-keeping and systematic cross-checks, and
- avoid data format “cul-de-sacs” (i.e. where data integrity and, most importantly, data relations are lost through, say, one-way aggregation).

The first point can be elaborated by saying “don’t over-complicate things”. It’s all too easy to over-complicate things, generating fresh code when available software functions can do the job.

note:

One can create extensive simulation-level SATURN networks in a spreadsheet with a little bit of effort. Global network changes can then be effected instantaneously!

3. AN UNUSUAL EXAMPLE OF EXTENDED SOFTWARE USE

General background

The specific example that has been chosen to illustrate an unusual application of multiple packages in a transport planning environment comprises the route set generation exercise undertaken as part of the development of the United Kingdom’s National Traffic Control Centre. The Traffic Control Centre is a real time traffic management system, initiated and owned by the Highways Agency. Currently, the Centre is still being implemented. In its final form, the Centre will provide real time traffic management and user information on the Highway Agency’s 15,000km road network. The rights to develop, implement and operate the system were won (through competitive bidding) by Serco, one of the world’s leading service companies. Traffic and broader consultancy expertise was and continues to be provided by Halcrow, a United Kingdom –based international consultancy.

It is precisely because the route set generation exercise featured the assignment and path building capabilities of Emme/2 (in combination with Emme/2's macro operation possibilities) that it provides an excellent case study for an Emme/2 users' group gathering. In so far as the route set generation exercise also involved substantial software-based data manipulation and GIS sifting downstream of the actual modelling the use of additional software packages is also illustrated.

Definitions

As stated, the example involved a route set generation exercise. By definition, and specifically for the purposes of the Traffic Control Centre design exercise, route sets were pairs of road sections, comprising a primary and an alternative or diversion route, between decision points on the project network and general destinations. A collection of route sets was called a library. On a practical level, the finalized route set library forms a part of the response plan generation module of the larger system (which includes option evaluation and information dissemination).

It needs to be stated that the development of a library of route sets formed only one part of what was an extremely large exercise involving, amongst other things, extensive software development, institutional planning and coordination, and the compilation of a comprehensive database-based description of the project network. In terms of its immediate context, the route set generation exercise formed a part of the following flow of work:

- building and populating of the project network in MapInfo
- additional validation of the MapInfo rendition of the project network with Saturn (validation checks were also undertaken in MapInfo itself)
- route set generation in Emme/2
- compilation of preliminary route sets using a custom-written computer program and MapInfo
- processing of the route set and project network databases in Access and Excel to add VMS (variable message sign) locations
- matrix generation and catchment factor (reflecting the traffic diversion potentials of the various route sets) calculation using Contram

The route set generation exercise was therefore not the only instance of multiple software use.

Actual process

The following outlines the methodology used to generate the library of route sets. Simply speaking, it involved three separate processes, namely:

- the automated generation of primary and secondary route sequences using standard transportation modelling software (i.e. Emme/2),
- the re-formatting, separation and consolidation of the primary and secondary route sequences into preliminary route set records using a stand alone and custom-written computer program, and
- the import of the preliminary route set records into a GIS (i.e. MapInfo) using a custom-written software interface such that finalized route set records and figures could be automatically assembled and visually sifted.

The table below summarizes these processes and their constituent steps. A brief description of the three processes and their steps is given in the pages that follow.

Processes and steps

process		step	
1	automated generation of primary and secondary route sequences using standard transportation modelling software	1	create model network
		2	generate matrix of possible project network movements
		3	validate network
		4	identify primary and secondary routes through repeated assignment
2	re-formatting, separation and consolidation of the primary and secondary route sequences into preliminary route set records using a stand alone computer program	5	re-format and separate primary and secondary route sequences generated in the assignment exercise to form individual route set records
		6	consolidate individual route set records according to common route set elements
		7	convert (1) "lead" route sequences into travel times and (2) node-defined sequences into link-defined sequences
3	import the preliminary route set records into a GIS using a custom-written software interface such that finalized route set database elements and figures can be automatically assembled and visually sifted	8	read records generated in the previous process and generate preliminary database of route sets
		9	automatically discard absurd or unlikely route sets
		10	visually scrutinize route set library and (1) discard absurd or unlikely route sets and (2) identify general destinations
		11	prepare a set of demonstration files and figures

Automated generation of primary and secondary route sequences using standard transportation modelling software

Step 1

Create an Emme/2 version of the project network

The Traffic Control Center project network database (in MapInfo) was used "as is" to create an Emme/2 version of the project network. Transfer of the network from MapInfo and the temporary renumbering of the nodes (for Emme/2 purposes only and because the node IDs were too long for Emme/2) was effected wholly by table transfers and Excel manipulations. The only additional effort required to produce a working network (i.e. an assignable network) involved (1) the flagging of mid-intercept links, (2) the addition of zones and zone connectors, and (3) the allocation of fixed but road class -dependent speeds to all links.

The zoning system chosen was hierarchical in nature, reflecting the general destinations (i.e. East Midlands, the North, etc.) associated with the project area with a finer but structured zonal breakdown within them. Zone connectors were generated at every (1) decision point and (2) major entry/exit point in the network. In the majority of cases decision points reflected single network junctions. There were a few cases, however, where decision points comprised several closely-spaced junctions or junction complexes.

Step 2

Generate a matrix of possible project network movements

Each O-D pair for the project area was inspected and its likelihood of using the project network to complete its movement was determined. Those O-D pairs that were unlikely to use the project network were excluded from the matrix of possible trips.

Step 3

Validate network

A series of preliminary assignments was undertaken to confirm the model's correctness (i.e. general connectivity and dimensioning) and suitability for purpose (i.e. sensitivity to alternate routings). Network coding and routing aberrations were then corrected with appropriate coding changes (network rendition errors were passed back to MapInfo).

Step 4

Identify primary and secondary route sequences through repeated assignment

This step actually comprised two distinct but parallel tasks, namely (1) Emme/2 assignments and (2) the (simultaneous) recording of primary and secondary route sequences. Each of these is dealt with in turn below.

Emme/2 assignments

- (1) An initial series of route set data, comprising a primary route and best alternate route for each O-D pair, was generated by:
 - (i) identifying minimum cost paths for each O-D pair, and
 - (ii) sequentially removing each flagged-link (potential "incident link") in the minimum cost path for each O-D pair and identifying the alternative used in each case, and
- (2) An extra series of route set data, associated with the same primary routes but comprising additional alternate routes for each O-D pair, was generated using a sequence of assignments where the primary and a growing set of additional alternative routes were rendered increasingly unattractive by reducing the travel costs on all unused portions of the network.

Recording route sequences

Every path identified in the Emme/2 assignments was explicitly recorded in terms of the link string it comprised, the initial link string it arose from (the minimum cost path or primary path), the link initially removed and the O-D pair involved. More specifically, the primary and alternate routes and their corresponding O-D pairs and “incident links” were written such that (1) the “incident link”, (2) the primary route sequence (in terms of a string of network nodes in sequence), (3) the secondary route sequence (also in terms of a string of network nodes in sequence) and (4) the origin and destination were explicitly recorded.

The recording exercise was undertaken via specially written Emme/2 macros which produced an ascii format file with a defined structure and content. The content (not the structure) is simply illustrated in the table below.

Field 1	Field 2	Field 3	Field 4
O-D pair 1111	primary route sequence 2222333344445555	incident link 6666	secondary route sequence 2222333377775555

Note: The numbers are purely symbolic, representing different data strings.

The notation for the primary and secondary route sequences was as follows:

common “lead” sequence	= “22223333”
decision link	= “3333”
common “tail” sequence	= “5555”
primary route sequence	= “4444”
secondary route sequence	= “7777”

Re-formatting, separation and consolidation of the primary and secondary route sequences into preliminary route set records

This occurred using a standalone and custom-written computer program which performed the following steps.

Step 5

Re-format and separate the primary and secondary route sequences to form individual route set records

The stand alone program read the ascii file generated from the Emme/2 assignments and:

- (1) identified the common and unique portions of the primary and secondary route sequences and also the decision link, and
- (2) wrote them all into separate fields.

The table overleaf illustrates the record contents.

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8
record number	common "lead" sequence	primary route sequence	secondary route sequence	common "tail" sequence	incident link	O-D pair	decision link
0000	22223333	4444	7777	5555	6666	1111	3333

Step 6

Consolidate individual route set records according to common route set (primary and secondary) sequences

The program then matched records with identical route set elements (i.e. identical primary and secondary route sequences) and grouped them accordingly. The unique relation between O-D pairs and their "lead" and "tail" portions was maintained in the process.

The table below illustrates the record contents.

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8	Field 9
old record numbers	new record number	common "lead" sequences	primary route sequence	secondary route sequence	common "tail" sequences	incident links	O-D pairs	decision link
0000a	0000b	22223333	4444	7777	5555	6666	1111	3333
0000a		22223333			5555	6666	1111	
0000a		22223333			5555	6666	1111	
0000a		22223333			5555	6666	1111	
0000a		22223333			5555	6666	1111	
.....		

Note: The unique relation between O-D pairs and their "lead" and "tail" sequences was maintained.

Step 7

Convert "lead" route sequences into travel times and node-defined sequences into link-defined sequences

Finally, the program:

- (1) converted the common "lead" sequences into single travel times (via lookup tables); and
- (2) converted the node-defined sequences into link-defined sequences.

The table below illustrates the modified record contents.

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6	Field 7	Field 8	Field 9
old record numbers	new record number	common "lead" times	primary route sequence	secondary route sequence	common "tail" sequences	incident links	O-D pairs	decision link
0000a	0000b	8888	4444	7777	5555	6666	1111	3333
0000a		8888			5555	6666	1111	
0000a		8888			5555	6666	1111	
0000a		8888			5555	6666	1111	
0000a		8888			5555	6666	1111	
.....		

Note: The unique relation between O-D pairs and their "lead" and "tail" elements was maintained.

Import the preliminary route set records into a GIS to assemble the finalized route set database and figures

Step 8

Read the records generated in the previous process and generate preliminary database of route sets

A custom-written interface program read select elements of the route set records generated in the previous process and created a route set database in MapInfo.

The basic elements of the database are illustrated below.

Field 1	Field 2 with cross-table link	Field 3 with cross-table link	Field 4 with cross-table link	Field 5 with cross-table link	Field 6
record number	common "lead" times	primary route sequence	secondary route sequence	O-D pairs	decision link
0000b					3333
	8888	4444	7777	1111	
	8888			1111	
	8888			1111	
	8888			1111	
	8888			1111	
	

Note: The relation between O-D pairs and their "lead" times was maintained.

Step 9**Automatically discard absurd or unlikely route sets**

Via cross-table lookups (within the system database itself), unreasonably long alternate routes (compared to their primaries) were identified and discarded (e.g. by setting an upper threshold to the ratio between primary and alternative route lengths).

Step 10**Visually scrutinize the route set library and (1) discard absurd or unlikely route sets and (2) identify general destinations**

The custom-written MapInfo interface not only created the required route set database structure but also visual representations (figures) of the route sets (in web-page format). These were visually sifted to further rationalize the library of route sets in terms of their basic practicality or sense. Similar route sets were also assessed to determine whether they could be resolved into single route sets. Further, destination patterns were identified and recorded in terms of general destinations.

Step 11**Prepare a set of route set files and figures for assessment**

Finally, a finalized set of route set files and accompanying figures were prepared.

Concluding remarks

It must be acknowledged that the route set generation exercise was easier to conceptualize than implement. Further, a vast amount of material (files, route definitions, etc.) was generated. The sifting of records to identify unique and common route sets, while preserving all contributory information (i.e. affected O-D pairs and their “lead” sections), demanded carefully written software and database operations and also a lot of visual sifting. However, the automation of the process, particularly the earlier steps, substantially reduced human involvement.

In retrospect, the approach chosen to generate the library of route sets proved to be something of an “over-kill” – in the cumulative sense, but not necessarily in terms of the independent elements of the exercise. That, however easily seen in hindsight, was not appreciated at the time. Neither does it negate the benefits, felt both at the time and downstream, derived from attempting it. The mere exercise proved useful to the project team, if only because the procedures and interfaces used are valid in principle to a broad range of planning applications. In terms of the integrated or complementary use of multiple software packages (MapInfo, Emme/2, Access, Excel and basic web-page programming) useful experience was gained.

4. ACKNOWLEDGEMENTS

The presenter's involvement with the route set generation exercise occurred during a year's working sabbatical in the United Kingdom with Halcrow, the consultancy responsible for providing traffic and broader consultancy expertise to the developer, implementer and operator of the system – namely, Serco. Recognizing both this fact and the broader context of the project, acknowledgement must be given to:

- David Turner and Tony Sharpe of Halcrow, London
- Dan Jones, an independent transportation modeller
- Serco, the company that won the right (through competitive bidding) to develop, implement and operate the National Traffic Control Centre
- the Highways Agency, the owner of the National Traffic Control Centre