Improved quality with better user interface in transport models?

Olav Kåre Malmin, olav.k.malmin@sintef.no
SINTEF Technology and society

Abstract

This study discusses the challenge transport analysts face when using the Norwegian Regional Transport Model (RTM) and it's user interface. The aim of this study is to identify typical challenges the users face when using a transport model like RTM, and discuss possible changes in future development of the transport model for better usability.

A review of the user support issues from 2013 to 2014 (N=62) shows that 63 % of all requests were related to the methods and procedures in the model and uncertainty on how to approach those methods in a analysis. 26 % of the requests were caused by direct and systematic errors in the input data.

The conclusion of the study is that the models user interface should be developed to help the user better understand the methodology of the transport model.

Introduction

The Norwegian Regional Transport Model (RTM, Tørset et al. (2008)) has been developed by SINTEF in a ongoing project since 2003. During this time SINTEF has also been responsible for user support. During the development work of the model, the experience from user support has lead to an improvement in the user interface to remove the most common problems users might have. This study of transport models user interfaces looks at what issues the users have had the last couple of years and discussed how to reduce those issues.

The aim of this study is to identify typical challenges the users face when using a transport model like RTM, and discuss possible changes in future development of the transport model to improve usability.

The evolution of user interface in Norwegian transport models

In the last 15 years, the transport models in Norway have evolved from very cryptic MSDOS-based models to a more user friendly model design in the Windows application Cube. At the same time, the way transport models worked changed from one specific model setup for each model area to a generic design, where the same model setup is used on different areas.

In the late 1990's, every transport model were developed and estimated for a specific city or region in Norway. Parameters for the demand model were more or less hard coded into the model. Model developer

and model user were usually the same team, so usability was not a concern. With the introduction of the regional transport model (RTM) in Norway from 2003 Tørset et al. (2008)) this changed so there was one transport model independent of the model area. The demand model (Rekdal et al., 2012) needed to be estimated and calibrated for each model area, and the user need to use the correct set of parameters for each study.

The early models were set up with more or less hard coded input and result files. To run different scenarios the entire model needed to be copied to a different directory and necessary files used to describe the scenario needed to be changed, or the file names for input files could be pointed at a different set of input files. This approach to scenario management required no user interface at all, but the user needed to have a broad insight into how the transport model works. This became increasingly difficult as the models grew more complex and important input files was scattered in different places in the models flow chart. The TASS3-model (Skjetne et al., 2000) was developed as a generic transport model but had to be customized for each city. This was the last transport model with no user interface and each scenario was a copy of the entire model.

The next version of the TASS-model framwork, TASS4 (Skjetne et al., 2003), and later TASS5 (Meland et al., 2006) took advantage of the new scenario manager offered in the Citilabs Cube suite. With the scenario manager it was possible to run several scenarios without copying the model. 1 shows the user interface for the TASS4 model. The user interface enabled the user to link scenario specific files for networks, public transit routes, zone data and various toll costs for each scenario in a tree view. The user interface relieved the user from having to copy the model for each scenario and change hard linked files. The more or less cryptic contents of the files was not changed. If the files contained an error the model would stop returning an error message.

Vegnett	C:\Diverse\modeller\Base\VNET.DAT	Browse ...	Edit ...
Koll lav	C:\Diverse\modeller\Base\KLINL.DAT	Browse ...	Edit ...
Koll hoy	C:\Diverse\modeller\Base\KLINR.DAT	Browse ...	Edit ...
Koordinater	C:\Diverse\modeller\Base\XY-TASS.DAT	Browse ...	Edit ...
Vegavgifter 1	C:\Diverse\modeller\Base\BOM0-1.DAT	Browse ...	Edit ...
Vegavgifter 2	C:\Diverse\modeller\Base\BOM0-2.DAT	Browse ...	Edit ...
Vegavgifter 3	C:\Diverse\modeller\Base\BOM0-3.DAT	Browse ...	Edit ...
Vegavgifter 4	C:\Diverse\modeller\Base\BOM0-4.DAT	Browse ...	Edit ...
Sonedata	C:\Diverse\modeller\Base\NYSONEDATA2003.DAT	Browse ...	Edit ...

Figure 1: User Interface of the Tass4 model

The RTM model developed from 2003 and the same model was used for any city or any region. This required the user interface to be more flexible. The users did not only have to link the correct files for each scenario, but also define which set of calibrated parameters for the demand model for each region. To reduce the number of files in the scenario manager, the parameters needed to be located in a specific directory for each region. The directory name needed to be the same as the name of region in the user interface. This lead to a confusing user experience where the user not only needed to define proper scenarios and edit the necessary files, but also manage a lot of files in the correct directory. Using input data files with pre-defined file names and path outside the scenario tree proved to be the source of many mistakes. If the files were present the model would run, but there was no guarantee the files were the correct ones for that region.

When the next version of RTM was released, the entire model was converted from the legacy TRIPS package to the script based Voyager package. TRIPS used a rigid input data format, while Voyager based models are very flexible. During the latest developments of the RTM-model the user interface could focus more on the users need to put correct data into the model, and not the user having to struggle getting the input data files correct in terms of file format. The flexibility of the Voyager package enables the model developers to define input data format and input methods without constrains from legacy software. There are however in the current RTM model some file formats and network attributes that are still defined because of constraints in the TRIPS package.

Other Norwegian transport models

There are other available transport models than the TASS and RTM model systems for the Cube software. Most importantly there are several EMME/3-based models using the same demand model as RTM, Tramod_By. The most commonly used model in the RTM23+ model for Oslo and surrounding municipalities (Rekdal & Larsen, 2008). These models use EMME/3 as network scenario management and result analyses, but the run flow is controlled by EMME-macros and batch files, which in turn is controlled by a variety of text files containing file names and various parameters. It's the end user's responsibility that all the control files are correct, and this is a very difficult task for everyone but the most experienced users of that particular model. There user group for this model is smaller and more concentrated than the RTM user group.

The user interface of the Norwegian RTM model

Input files

The user interface of the RTM model (Malmin, 2013) is managed by Citilabs Cube Base. The core of the user interface is a scenario manager. The scenario manager organizes the scnearios in a hierarchic tree. The default tree, shown in figure 2 consists of the model area of each administrative region on top and then various forecast years. The user is free to choose whatever scenario tree that seems fit for each analyses.
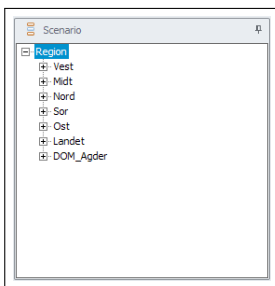

Figure 2: The scenario manager of Cube Base

In addition to the scenario manager, Cube Base manages inputs files and options for the scenario. The user input is defined independent of the scenario tree, which means that each scenario require exactly the same files and options. Input data to the RTM model, shown in figure 3 are 27 different files divided into 12 groups of unique file formats. Not all files are required, and the requirement depends on type of analysis and model area.

The most important main group of files are:

Geodatabase The geodatabase contains networks and public transport routes.

Additional costs Public transport costs, internal distances and toll zones.

Fixed trip matrices Heavy weight vehicles and external trips.

Zone data Land use, employment, education and demographics.

Parameters Various parameters and premises for the demand model.

Traffic count Traffic count data for validation.

System complexity

(**?**, p. 153-155) describes the challenges about an increasing number of computer systems to perform various tasks. A computer system means components like operating systems, software, or interactive websites like a database front end. Using the RTM transport model requires knowledge of several computer systems. To be able to use the model and perform credible analyses, the users needs to be skilled in all the systems. If the user are challenged in one of the systems, the results might be wrong, difficult to explain or both. Time consumption will also be a factor.

Figure 3: First page of the scenario editing in the RTM model

The transport models requires the following systems:

Microsoft Windows Explorer Most user have a certain skill using MS Windows. Using transport models requires the user to be able to browse complex directory structures in Explorer to locate both input and result files. It is also important during installation of the transport model that all files are put into sensible directory names.

Citilabs Cube The transport model is developed for use in the Cube transport planning software suite. The user interface, execution of scenarios and result analyses are handled in Cube. The user needs a understanding of how the Cube software works and it's limitations.

ESRI ArcGIS ArcGIS is used for creation and editing of input networks, public transit routes and zone data. A extension for ArcGIS is used for network editing. Using ArcGIS with the transport network extension (TNExt) is the most challenging aspects of using the RTM model system. The used both need experience with ArcGIS and TNExt. And on top of that, the user need experience to create network and public transit routes that makes sense in a transport model analysis.

DBF editor Various input files are database tables in DBF-format. Microsoft Excel 2010 and later do not support DBF-files. An alternative is OpenOffice Calc with proper support for DBF-files. Because of the lack of support in MS Excel, there have been some confusion creating and editing DBF-files for the RTM model.

Text editor A decreasing number of input files for the model are still raw text files. Surprisingly, editing a simple text file can lead to errors if the model expect tab-separation or similar. Character set (ISO, Unicode) are also confusing in Nordic and other European countries.

To set up a single scenario run in the transport model, the user needs to use four different software packages to create 12 groups of a total of 27 input files. This complexity puts a very high demand on the user. Söderström (2013, p. 154) describes that the theoretical relations between systems increase drastically when more systems are added to the complexity. 4 systems give 4 possible relations, while 5, 6 and 7 leads to 10,15 and 21 relations respectively. The probability of error increases with number of relations.

Error messages and feedback

There are two different types of errors:

Direct errors in the input data like values outside the defined limits and wrong data format. This type of error will stop the execution of the program with a error message.

Systematic errors will not stop any program execution but will give bad results from the model. Input data networks with wrong attributes or incorrect one way drive directions are typical. A good user interface must aim to guide the user so the risk of systematic errors are reduced.

The various systems returns different error messages and feedback. The RTM model will on a successful run create a scenario raport (PDF-file) with information about the scenario's run details, input data and key results. This report was developed in the RTM model so it would be easier for the user to keep a record of a scenario run for project documentation and quality assurance. In this report, there are a few logical tests performed on the input data and the user needs to evaluate the results of these tests. This helps reducing systematic errors. Possible systematic errors are asymmetry in the level of service data because of network errors and unconnected zones. The model will run just fine with these errors, but the results will be skewed. There is also a summary of all the input parameter files to the demand model. If the user have provided a insufficient number of files, or given the files the wrong file names, the scenario report will highlight possible mismatches.

The scenario report contains warnings about possible errors in the input data are not always evaluated by the user. There have also been users that have failed to interpret the warnings. The scenario report also contains a list over all input files used by the scenario, but as long as all the different file types are linked to the correct file dialog box in the user interface and the model does not abort, it is not possible for a computer program to interpret if a file is correct or not. The user must have some sort of quality control to make sure all files in the user interface are linked correctly.

The transport network editing extension in ArcGIS (TNExt) will give error messages for actions not allowed in the GIS environment and will also give error messages for mismatches between the various data layers. The system will not however indicate if something is wrong with the network or public transit lines. The results of such errors will materialize in the logical tests in the PDF-report after a model run or during manual inspection of the result networks from the transport model.

The most common error messages appear in Cube software that organizes scenarios and run the model. In the user interface where all the files, options and parameters for a scenario are defined, only file existence are checked. It is up to the transport model developer to code routines to check the validity of the input data. In RTM this is not necessarily the case. If the user have linked the wrong type of files, for example a database file where a network file should be, the model will eventually abort when the file in question is accessed by the transport model.

The error messages reported by Cube Voyager when the model aborts are highly specific on what caused the error. But to figure out what exactly is wrong from the error message requires experience, experience the user does not necessarily have. The error messages makes sense for a model developer, but most users have not developed transport models. The challenge from a user support perspective is how to communicate those error messages. Users have asked for help with errors with a variety of detail, from a simple screenshot of the abort message to proper log files. Usually a lot of issues can be solved by users providing log files. But in some cases log files do not lead to a solution and then parts of, or the entire model, has to be investigated. This is quite easy with various desktop sharing tools like Lync or TeamViewer, but earlier all the files needed to be transferred electronically or physically.

Common user challenges

There are generally two different groups of users of a transport model. The first group is the consultant who works on projects where the model is used to analyse different scenarios like a road development project or public transport improvement. This user group have time constraints and rely on a working model and that establishing the scenarios in the model is moderately easy. When issues arise, there is little time to figure out what went wrong.

The second user group is the user who are more skilled in model development. When a issue arise, these users figure out most issues themselves. But there have been situations where users in this group have fixed problems by tinkering with the model until the problem is solved, and not figured out what in the input data that caused the issue in the first place. Tinkering with the model to solve issues are not a desirable procedure since it will be more difficult to recreate the results later.

Requests for user support can be divided into five categories:

Input error There was an error in one of the input files. This will lead to wrong results or aborted model run.

Scenario error The scenario was not created correctly. This usually leads to an aborted model run.

Confusion The user did not understand methods of preparing input data or the model's premises.

Methodology The user asked questions to better understand the model.

Bug report The user discovered something wrong with the model.

All e-mails, telephone calls and instant massages requesting user support are logged since May 2014. Before 2014, only e-mails are logged. In this study all logged user requests for 2013 and 2014 were counted and categorized, shown in table 1.

Table 1: User support requests in various categories

Category	N	Share (%)
Input error	6	10
Scenario error	10	16
Confusion	12	19
Methodology	27	44
Bug report	7	11
All requests	62	100

The table shows the complexity of developing a good user interface for a transport model. Improving the user interface for the model itself only helps reducing the scenario errors, while the other categories needs attention to the other systems used to create input data. The study shows that the majority of the user support requests were questions about the models methodology and the users also misunderstood how the model worked (63% of all requests). The great challenge for model developers is to create a user interface that improves understanding and reduces confusion. Educating the users in both using and preparing data for the model is also essential. (Söderström, 2013, p. 93).

10% of the users had problems preparing the input data correctly. The difficulties with transport models are the indirect input data method where the users first needs to prepare all input data, and then tell the model where to find the data.

Not all users ask user support if they are uncertain about aspects of the model. There have been several issues where the model was used on numerous false premises, and where the results were presented in a way that his all the systematic errors. This happens because the model in it's current version allows wrong use. To counter this problem, the model needs to perform more self tests that feeds back to the user.

Resources

A transport analysis project in Norway have three phases:

1. Data collection

2. Establish scenarios

3. Run scenarios

4. Cost benefit

5. Analyses and reports

In a typical project, data collection and scenarios takes half the allotted time, running the model for all scenarios doesn't require man hours, but the computers can run for days to weeks. In an ideal world the results should be analysed and reported, but it's quite common that the analyses reveal some systematic error or bad scenario design. Then it is back to step 2 and the resources disappear fast. The commissioner

wants the job done with less cost, and extra steps will cost more in either monetary units or quality. If the model's user interface reduced the risks of error, there would be less wasted resources. The challenge is to spend these saved resources on better scenario design and more clever analysis.

Usability of transport models

Most transport models have been developed with strong focus on features and valid results. The study in the previous section shows that users have challenges with a transport model like RTM that were developed with usability in mind. This shows that the model's user interface is still not good enough. The challenge is to create a user interface that helps the user understand the methodology of the model, since these issues are most in demand. Since a transport model is not a stand alone product, but dependent on different software to create input data, it is necessary to evaluate the whole eco-system. If it is not possible to change the software that creates parts of the input data, how the input data is presented to the model can be changed for better work flow.

Söderström (2013) and Gould (1995) describes the importance of not only test the software on a user group, but actually involve users in the development process. This way the users will get a better product, but also get ownership to the model. This type of development require quite a change of current procedures, but the result will lead to a transport model that is easier to use with a lesser risk of creating errors.

Discussion and conclusion

This study argues that a transport model needs a user interface that guides the user to better understand the methodology and premises of the model. This will in turn lead to results with less systematic errors. The user have to use several software packages to establish input data to scenarios and analyse results. There is a risk of data produced by other systems will have wrong contents, format or both. This must be taken into consideration when developing models.

44 % of the queries from users were questions about methodology in the initial phase of the transport analysis. This shows that most users are willing to learn the system, but use user support to do it. There is clearly a demand for more training courses in not only using the transport model, but also prepare input data.

References

Gould, John D. (1995): *Human-computer interaction*. chapter How to Design Usable Systems, pages 93–121, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ISBN 1-55860-246-1, URL `http://dl.acm.org/citation.cfm?id=212925.212935`.

Malmin, Olav Kåre (2013): *Cube-teknisk dokumentasjon av regional persontransportmodell versjon 3*. SINTEF-rapport A24718, SINTEF Teknologi og samfunn, Trondheim.

Meland, Solveig, Eirik Skjetne, Trude Tørset & Olav Kåre Malmin (2006): *TASS5 for Trondheim*. SINTEF-rapport A3196, SINTEF Teknologi og samfunn, Trondheim.

Rekdal, Jens & Odd I. Larsen (2008): *RTM23+ Regional modell for Oslo-omrdådet*. Rapport 0806, Møreforsking Molde AS, Molde.

Rekdal, Jens, Odd I. Larsen, Arne Løkketangen & Tom N. Hamre (2012): *TraMod_By Del 1: Etablering av nytt modellsystem*. Rapport 1203, Møreforsking Molde AS, Molde.

Skjetne, Eirik, Jorun Gjære, Kristin Kråkenes & Børge Bang (2000): *TASS. Transportanalysemodell for strategiske studier - versjon 3.1*. Rapport, Scandiaconsult AS.

Skjetne, Eirik, Trude Tørset & Olav Kåre Malmin (2003): *TASS Trondheim, versjon 4.0. Transportanalysemodell for strategiske studier*. SINTEF-rapport STF22 A3320, SINTEF Bygg og miljø, Trondheim.

Söderström, Jonas (2013): *Jævla drittsystem! Hvordan IT-systemer kan ødelegge arbeidsdagen og hvordan vi kan ta tilbake kontrollen*. Spartacus Forlag, ISBN 978-82-430-0797-0.

Tørset, Trude (2013): *Cube-regional persontransportmodell versjon 3*. SINTEF-rapport A24717, SINTEF Teknologi og samfunn, Trondheim.

Tørset, Trude, Olav Kåre Malmin, Snorre Ness, Ina Abrahamsen & Oskar Kleven (2008): *Regionale modeller for persontransport. Modellbeskrivelse*. SINTEF-rapport A3973, SINTEF Teknologi og samfunn, Trondheim.