Arbeitspapier

179

Ahmed Talaat, Björn Schwarze

The Dortmund Region Networks
Scenario Generation Module

This working paper is originated from
the PROPOLIS and ILUMASS research projects.

Dortmund, May 2003
Table of Contents

Introduction ........................................................................................................................................ 4

1. Rationale for a Networks Scenario Generation Module .......................................................... 7
   1.1 Supporting Various Data Representations ......................................................................... 7
   1.2 Handling Different Temporal States of the Network ..................................................... 8
   1.3 Complicated Coding System ......................................................................................... 9
   1.4 Automated Editing and Displaying Processes ............................................................. 10
   1.5 Development of Scenarios ......................................................................................... 10

2. Representations of Networks in ArcInfo ................................................................................. 11
   2.1 The Arc Attribute Table ............................................................................................. 11
   2.2 The Links Data Table ............................................................................................... 12
   2.3 The Route Attribute Tables ....................................................................................... 14
   2.4 The Node Attribute Table ........................................................................................ 17
   2.5 The Turn Off Restrictions Table ................................................................................ 17

3. Export of Networks ............................................................................................................. 19
   3.1 Export of Networks to the Dortmund Model ............................................................... 19
      3.1.1 Link Records of ZVMxx.DAT ........................................................................... 19
      3.1.2 Line Records of ZVMxx.DAT ........................................................................ 21
   3.2 Export of Networks to the PROPOLIS-Raster Module ................................................ 23
      3.2.1 Link Attributes File (NLDO.DAT) .................................................................. 23
      3.2.2 Link Coordinates File (NLDO.LIN) ................................................................ 25
      3.2.3 Node Attributes File (NNDO.DAT) ................................................................ 25
      3.2.4 Node Coordinates File (NNDO.PTS) ............................................................. 26
   3.3 Export of Networks to the ILUMASS Model ............................................................... 26
      3.3.1 Link Records of TR_xx_yyyy.DAT ................................................................. 27
      3.3.2 Line Records of TR_xx_yyyy.DAT ................................................................ 30
      3.3.2 Links Geometry File (TR_xx_yyyy.LIN) ........................................................ 32

4. The Networks Module Interface – User Guide ..................................................................... 33
   4.1 The Main Window ....................................................................................................... 33
   4.2 The Display Submodule ............................................................................................... 34
   4.3 The Edit Submodule .................................................................................................... 36
      4.3.1 Setting Initial Parameters ............................................................................... 36
      4.3.2 The Selection Tools ......................................................................................... 37
      4.3.3 Updating Attributes ....................................................................................... 37
      4.3.4 Editing Links and Public Transport Lines ....................................................... 38
   4.4 The Output Submodule ............................................................................................... 40
5. Technical Description – Developer Guide ................................................................. 42
  5.1 Technical Specifications ......................................................................................... 42
    5.1.1 Software .......................................................................................................... 42
    5.1.2 Hardware .......................................................................................................... 42
    5.1.3 Input Data ......................................................................................................... 42
    5.1.4 Output Data ...................................................................................................... 43
    5.1.5 Distribution ...................................................................................................... 43
  5.2 Auxiliary Input Data .............................................................................................. 45
    5.2.1 Coverages ......................................................................................................... 45
    5.2.2 Info Tables ....................................................................................................... 46
    5.2.3 Symbol Sets ...................................................................................................... 46
    5.2.4 ASCII Files ...................................................................................................... 46
  5.3 System Structure ................................................................................................... 47
  5.4 Technical Description ........................................................................................... 48
    5.4.1 SSADM Terminology ...................................................................................... 48
    5.4.2 Overall System Description ............................................................................. 48
    5.4.3 Detailed System Description ............................................................................ 50

6. Future Developments and Enhancements ................................................................. 80
  6.1 Enhancements to the Display Submodule ............................................................... 80
    6.1.1 Different Symbolization to Different Link Categories .................................... 80
    6.1.2 Map Composition and Printing Current Screen ............................................. 80
  6.2 Enhancements to the Edit Submodule .................................................................... 80
    6.2.1 Adding Node/Splitting Link ........................................................................... 80
    6.2.2 Attribute Names Instead of Attribute Codes ................................................... 81
    6.2.3 Selecting Nodes by Station Name .................................................................... 81
    6.2.4 Selecting Routes by End Stations .................................................................... 81
    6.2.5 Integrating the Routes Restoration Scheme .................................................... 81
    6.2.6 Scenario Manager ............................................................................................ 81
  6.3 Enhancements to the Output Submodule ............................................................... 82
    6.3.1 Error Detection Schemes ................................................................................. 82
    6.3.2 Exporting Land-Use Data ................................................................................ 82
    6.3.3 Reference Year and Scenario Number for Export to ILUMASS ..................... 82
  6.4 General Enhancements .......................................................................................... 83

References ..................................................................................................................... 84
**Introduction**

Spatial models have become an important branch of scientific endeavour. They are commonly utilized in a wide range of applications to simulate real world activities so that the future behaviour of the world they represent can be revealed. Integrated transport and urban land-use planning models in particular have been developed and adopted by the Institute of Spatial Planning, University of Dortmund (IRPUD) in several research projects with the overall objective of studying how the changes of land use and the changes in transport infrastructure would affect each other and the environment. Transport models simulate passengers travel and good transport based on submodels of flow generation, destination choice, modal choice, network search and flow assignment. In more recent developments, concepts of activity-based mobility have been taken up by transport modellers to take account of changes of land-use and the resulted effect on transport demand (Wegener, 2000a).

As such models are increasingly covering a large geographical area with a detailed representation of the transport networks or land-use polygons, the use and utilization of Geographic Information Systems (GIS) has been introduced to help modellers handle the wealth of information involved. GIS software packages are used mainly as a tool for managing spatial information and performing associated computations. Their database management features offer an effective manipulation and representation of spatial and attribute data. This is particularly suited to the kind of work carried out using a land-use/transport model (Sheppard et al., 2000). A GIS could dramatically increase the speed at which data processing can occur and the ease with which results can be analysed and displayed.

However, the lack of modelling functions in GIS and the difficulties of linking external models to GIS have resulted in a considerable gap in the integration between GIS and spatial models (Wegener, 2000a). As this gap continues to hinder the full utilization of GIS in transport planning, research efforts have been exerted at IRPUD during the last decade to bridge this gap. Under the umbrella of different research projects, several interfaces were developed at IRPUD to link GIS to commercial as well as to in-house transport models. An interface that contains a set of sophisticated editing tools was developed to link EMME/2 (INRO Consultants, Montreal, Quebec, Canada) and ArcInfo (Environmental Systems Research Institute, Redlands, CA, USA) for the TRILAT project (Schauerte-Lüke et al., 1998) as the first step in such efforts. This interface contains macros for network aggregation and network data transfer as well as a network editor for adding, altering and adapting attributes in ArcInfo. In a further step, an interface was developed to link ArcInfo to the commercial transport model VSS (Harlof Hensel Stadtplanung Ingenieur GmbH, Aachen, Germany) and to the Raster-Module (Spiekermann & Wegener Stadt- und Regionalforschung, Dortmund, Germany) for the VuGIS project (Schürmann and Schwarze, 2002). This comprehensive interface, which was developed to manage the conversion and transfer of the complicated data involved among these systems, can either fully be inserted in the integrative VuGIS system or alternatively can be interactively activated by the user. In a similar attempt, an interface was developed to extract the strategic networks from the base networks and to transform ArcInfo binary data into the ASCII input format required by the SASI model (Wegener et al., 2000).

The Dortmund Region Networks Scenario Generation Module (or the Networks Module for short) is the most recent interface developed at IRPUD in this respect. It is mainly intended to provide the necessary tools to display and edit the Dortmund region transport networks. In addition it is intended to export both the attributes and geometry of the networks from the
GIS-based database to three different external models from the fields of transport, land use and environment. These are the Dortmund Model, the PROPOLIS-Raster Module and the ILUMASS Model. With these capabilities, the Networks Module confirms the line of interfaces developed at IRPUD to closely link GIS, transport, land-use and environmental models.

The Networks Module was initially designed and developed in the framework of the PROPOLIS project (Planning and Research of Policies for Land Use and Transport for Increasing Urban Sustainability). Objectives of the project are the further development of theories about the interaction between transport and spatial development, the improvement of planning, simulation and evaluation methods and the identification of planning policies for making urban regions in Europe more sustainable (LT Consultants Ltd, 2003; Lautso et al., 2002). To achieve these objectives, existing land-use transport models of seven European urban regions (Bilbao, Brussels, Dortmund, Helsinki, Inverness, Naples, Vicenza) are enhanced and complemented to enable them to forecast the impacts of different policies on spatial development and indicators for the evaluation of their effects on sustainability. The task of IRPUD in PROPOLIS is to analyse the relationship between environmental quality and location behaviour, to refine and complement spatially disaggregate environmental and location models and to apply the urban simulation model (the Dortmund Model) developed at IRPUD to the urban region of Dortmund for the evaluation of a common set of policies to be examined.

Two main components make a great use of the Networks Module: the Dortmund Model and the PROPOLIS-Raster Module. The Dortmund Model is a simulation model of intraregional location and mobility decisions in a metropolitan area (Wegener, 2000b). It receives its spatial dimension by the subdivision of the study area into zones connected with each other by transport networks containing the most important links of the public transport and road networks coded as an integrated, multimodal network including all past and future network changes. It receives its temporal dimension by the subdivision of time into periods of one or more years' duration. The model predicts for each simulation period intraregional location decisions of industry, residential developers and households, the resulting migration and travel patterns, construction activity and land-use development and the impacts of public policies in the fields of industrial development, housing, public facilities and transport. The Networks Module exports the road network as well as the public transport lines to the Dortmund Model in a specific ASCII file.

The PROPOLIS-Raster Module calculates indicators in a wider modelling framework for which a disaggregate treatment of space is required (Spiekermann, 2003a; 2003b). It uses raster techniques to disaggregate zonal data to raster cells for the calculation of spatially disaggregate environmental and social indicators such as transport emissions, quality of open space, exposure to poor air quality or exposure to noise. The Networks Module is used to export the network links and nodes to ASCII files as inputs to the PROPOLIS-Raster Module. The main difference to the export to the Dortmund Model is that the Raster Module requires also the exact alignment of the transport links to be capable to localise the precise locations of emissions and to calculate proper land fragmentation indicators.

In a later stage, the Networks Module was modified to support the transport model of the ILUMASS project. The project ILUMASS (Integrated Land-Use Modelling and Transportation System Simulation) aims at embedding an existing microscopic dynamic simulation model of urban road traffic flows into a comprehensive model system incorporating both changes of land use and the resulting changes in transport demand. Study region for tests and
first applications of the model is the urban region of Dortmund. The common land-use and transport database is compiled by IRPUD in co-operation with the City of Dortmund. In this respect, the Networks Module is the main tool used for the preparation and production of the transport database to the ILUMASS project.

The study area in both the PROPOLIS and ILUMASS projects, and hence the main focus of the Networks Module, is the metropolitan area of Dortmund. This comprises the commuter catchment area of Dortmund containing Dortmund itself and eighteen neighbouring communities. The study area is referred to throughout this document as the Dortmund region.

This working paper presents a detailed technical description of the Networks Module and its components as well as the underlying structure of the networks data in both the GIS-database and the output ASCII files. The document is intended for the users of the Networks Module who will find all the details they need to understand and use the system. It is also intended for system developers and programmers who may need to modify and enhance the functions of the Networks Module.

After this introduction, a further description of the objectives and purpose of the Networks module is presented in Chapter 1 “Rationale for a Network Scenario Generation Module”. Then the representation of networks in the spatial database within ArcInfo is described in Chapter 2 “Representations of Networks in ArcInfo“. The description of the output files produced by the Networks Module to each of the external models follows in Chapter 3 “Export of Networks”. A simplified user guide that explains how to use the system interface is presented in Chapter 4 “The Networks Module Interface – User Guide“ and the detailed technical description of system components and structure is found in Chapter 5 “Technical Description – Developer Guide“. Finally, a set of suggestions and recommendations for the future development of the Networks Module is listed in Chapter 6 “Future Developments and Enhancements”.
1. Rationale for a Networks Scenario Generation Module

As described in deliverables D5.1 of the PROPOLIS project (Sheppard et al., 2000), for the implementation of the Dortmund Model and the Raster Module in PROPOLIS, and also for the preparation of the database in ILUMASS, ArcInfo is used for efficient capture and maintenance of geometry data such as zone boundaries and land-use data as polygons (or area) data and transport network data (roads, railways, public transport lines) as vector (or line) data. On the other hand, the land-use transport model is used for efficient calculations on these data, such as the simulation of location behaviour of firms and households and the mobility behaviour of travellers for work, shopping, education and social trips.

Both functions, data capture/maintenance and modelling, have different requirements. The geometry data within the GIS contain a great amount of detail with respect to location and shape (alignment) of zone boundaries and transport links. This detail is not required for modelling where, for instance, zones are abstracted to areas and transport links to their length or travel time or cost. On the other hand, models require a temporal dimension, such as changes in land use or transport networks, which cannot be easily implemented in GIS.

For efficient model operations it is therefore necessary to convert GIS data to model input by removing irrelevant information and adding required information which is not – or only partly – contained in the GIS. This conversion is performed by the Dortmund Networks Scenario Generation Module (the Networks Module). The Networks Module can be seen as an interface between the GIS and the model by composing the input data required for a particular simulation run.

The Networks Module will therefore (a) extract relevant link information from the network representation in ArcInfo, (b) combine the information of the road and public transport network into one multimodal network representation, (c) convert the route representation of public transport lines to the equivalent representation accepted by the Dortmund and ILUMASS Models network representation and (d) incorporate the information about network changes from the ArcInfo ASCII export file to a network scenario file with multiple records for each link or line which undergoes changes.

Outputs of the Networks Module are ASCII files that contain either networks attribute or geometry as inputs to the three supported models: the Dortmund Model, the PROPOLIS-Raster Module and the ILUMASS Model.

Since all input geographic database is stored in ArcInfo, and since ArcInfo is virtually capable of performing most of the tasks required from editing to exporting the data, some questions may be raised: what is the rationale for the Networks Module? What is the need to develop an extra interface on top of ArcInfo? And what are the advantage of using this interface to simply using ArcInfo commands? This chapter is aimed at answering all these questions.

1.1 Supporting Various Data Representations

One of the main challenges addressed by the Networks Module is the support of three different external models: the Dortmund Model, the PROPOLIS-Raster Module and the ILUMASS Model. A detailed description of the data structure accepted by each model is available in
Chapter 2 “Export of Networks”. Although the three external entities require ASCII input files, the contents and the structure of these files differ. Therefore, output files produced by the Networks Module as inputs to these three external models can not only be produced by ArcInfo common commands such as Unload or Ungenerate. Some sort of customisation is required in order to produce each model-specific data format.

A set of AML macros were written to automate the export processes. These macros are called from within the Networks Module interface to produce the different output files without users interference. A detailed description of these macros and how they are integrated within the system is available in Section 5.4.2 “Overall System Description”.

1.2 Handling Different Temporal States of the Network

Because the Dortmund region transport network is used to calculate changing inter-zonal travel times over time, past and future development and different temporal states of the network have to be stored and manipulated within the database. However, it is hardly possible in GIS to represent the temporal evolution of networks, i.e. the addition, modification or deletion of links at certain points in time (Wegener, 2000a). Storing several data sets with different time labels is still a dilemma in common commercial GIS packages including ArcInfo.

Several approaches were explored to solve this problem in the Networks Module. One approach is to create a different ArcInfo network coverage for each time label. The clear disadvantage here is the redundancy in data that results from storing the same features several times in different coverages. Another problem is the implementation and maintenance of edit changes in the different coverages over the time. Therefore, this approach can only be accepted when time interval is fixed, when the time resolution is high, i.e. several years, and when the number of simulation years is limited. In other words, this approach can be applied when only a few number of coverages is required. This is not the case in the Dortmund and the ILUMASS models where the simulation period is one year and more than thirty simulation runs are desired.

Another approach is to store all sets of attributes over time in the arc attribute table in a horizontal manner so that all fields of attributes are repeated for each simulation year. In other words, the arc attribute table will have as many set of attribute fields as the number of simulation years. Again this results in a huge sized attribute table that can be too difficult to store, manage and retrieve. Moreover, it does not solve the redundancy problem because each arc may experience a maximum of two or three changes over the history. That means that most of the records will be assigned a null value under several fields of attributes of several years, but they will nevertheless occupy the complete cell size of the storage capacity.

The third approach is to introduce an external info table that stores multiple sets of attributes with different time labels for each arc feature in the network. This approach stores the records of attributes vertically rather than horizontally as in the previous approach and hence saves a large amount of disk space. Attributes will be recorded only when the arc experiences a certain change and therefore table size is kept to the minimum. This is the approach adopted by the Networks Module.
The external table is then linked to the original arc attribute table through a key unique identifier in a one-to-many relation. ArcInfo supports one-to-many relations only through the use of AML cursors (ESRI, 2002). Therefore a set of AML Macros were written and integrated in the Networks Module interface to allow the storage, modification and display of these temporal information.

While this is the case for the arc features, a certain degree of redundancy had to be compromised in the route features. Routes features are linear features that are composed of a series of links that represent the path of a certain public transport line. As mentioned before, changes in the network links is limited to the attributes describing each link, i.e. link category, maximum speed, type of change, etc. Route features in the contrary, have spatio-temporal changes. A transport line path may change as well as its attributes. Therefore a new route has to be established and stored for each change in the transport line.

1.3 Complicated Coding System

In addition to the wide range of time-based attributes that have to be stored and managed by the Networks Module, a relatively intensive coding system was developed to fulfill the requirements of the three supported external models. As described in Chapter 3 "Export of Networks", the codes and attributes accepted by each model differ. This results in a set of challenges for the users if they are left alone to edit and export the data with the traditional ArcInfo tools.

One challenge is the street category codes. The Dortmund Model has its own street links classification that differs to the one adopted by the PROPOLIS-Raster Module. The Dortmund Model has a more detailed category definition, and hence more categories, than the PROPOLIS-Raster Module. That means that the categories adopted by the Dortmund Model have to be aggregated to produce those adopted by the PROPOLIS-Raster Module. The translation between these two classifications is implemented in the Networks Module on-the-fly during the export process. Both classifications are explained in Section 3.2 "Export of Networks to the PROPOLIS-Raster Module".

Another challenge is the capture, storage and production of the turning restriction rules at intersections, which are considered by the ILUMASS Model. Such rules are partially defined based on general knowledge of the users of the study area, but mainly developed automatically by an AML macro developed for this purpose. Section 3.3 "Export of Networks to the ILUMASS Model" provides more details on this topic.

The most complicated task is to maintain the street code system that combines the link category and any transport line that may travel over this link. Looking after this code system, while editing arcs and routes using traditional editing tools, would certainly confuse the users. For example, if the path of a certain route is changed at a certain point of time, and accordingly that route leaves a certain link, the link code should be altered to reflect this change. In the contrary, if a certain link is deleted or modified, all routes that travel over this link should be modified accordingly. The Networks Module allows the user to edit the different features separately. This combination of codes is then calculated in a rather complicated process during the export process. Chapter 2 "Representations of Networks in ArcInfo" demonstrates how attributes of arc and route features are stored separately in ArcInfo, while Section 3.1
"Export of Networks to the Dortmund Model" provides more details on how the combined link code is exported to the Dortmund Model.

1.4 Automated Editing and Displaying Processes

The Networks Module contains several useful functions that automate some of the editing and display processes. The node numbering function, for example, automates the assignment of a valid number to newly added nodes. It composes the six-digit node number (see Section 2.4 "The Node Attribute Table") by concatenating the zone number where the node is located and the node next free sequential number available for this zone. This process involves two different analyses. The identification of the zone number involves an overlay analysis between the street network and the zones coverages while the identification of the next sequential node number involves statistical analysis over the nodes in the network. Both processes are performed on-the-fly. A new node number is assigned to the node and to the from- or to-node number of the corresponding arc without that the user realizes the underlying complexity.

Restricting the display to features that are built before a certain reference year is another useful function. While editing the user may like to focus on a certain point of time for which a paper map is available. Working with all features that are built before and after this time can be very confusing. This function only shows features built before a specific year and temporarily hide all other features.

1.5 Development of Scenarios

The development of different scenarios that represent certain changes in the networks is the main task of the Networks Module. The user can easily carry out certain changes that represent a certain scenario and export the whole network to the relevant model. Scenarios can be based on spatial changes, e.g. the deletion or addition in the street network or the transport lines. It can also be based on changes in the attributes such as the improvement of a certain part of the street network or the change of the maximum speed allowed. Regardless of the kind of change, the Networks Module can be easily used to develop and export as many scenarios as possible.
2. Representations of Networks in ArcInfo

The geometry of the transport networks of the Dortmund region is stored in an ArcInfo network coverage called STRECKEN. The coverage is composed of three types of features: arcs that represent links of roads and rail tracks, routes that represent public transportation lines, as well as nodes that represent roads intersections, bus stops or rail stations. Five different kinds of attribute tables are developed to store and describe the information associated with these features. These are the Arc Attribute Table (STRECKEN.AAT), the Node Attribute Table (STRECKEN.NAT), the Route Attribute Table (STRECKEN.RAT<subclass>), the Links Data Table (STRECKEN.DAT) and the Turn Off Restrictions Table (STRECKEN.TOR). The structure of these data elements within ArcInfo is represented in Figure 2.1.

A detailed description for each of these tables and their contents is provided below. This description focuses on items that store attributes relevant to the Dortmund Model, the PROPOLIS-Raster Module and the ILUMASS Model or those, which directly describe the database. It also describes items that support certain operations within the Networks Module. ArcInfo internal items are not described in this section.

2.1 The Arc Attribute Table

The STRECKEN.AAT stores basic information that describe links in the network coverage. For each link of the road network, information on link length, start-node, end-node, and a unique identification number are stored in this file (see Table 2.1).
Table 2.1 Attributes of the STRECKEN.AAT.

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Item Definition</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>VON</td>
<td>7, 7, Integer</td>
<td>From-node number</td>
</tr>
<tr>
<td>NACH</td>
<td>7, 7, Integer</td>
<td>To-node number</td>
</tr>
<tr>
<td>LINIE</td>
<td>4, 4, Integer</td>
<td>Impedance item</td>
</tr>
<tr>
<td>KNOTEN</td>
<td>14, 14, Integer</td>
<td>Redefined item representing both from- and to-node numbers</td>
</tr>
</tbody>
</table>

The from-node and to-node numbers accord with the Dortmund Model numbering system (see Node Attribute Table, Section 2.4). As a rule, the from-node number is always smaller than the to-node number.

The impedance item is used to set an impedance value for each link to support the process of building the route systems within the Route Restoration Scheme (see Section 5.4.3 “Detailed System Description”). The initial default value for all links is set to 1000.

The redefined item is composed of both the from-node and to-node numbers. It is unique for each link because each two nodes in the network are connected by one link only. This item is used as the key link between the STRECKEN.AAT on one side and the STRECKEN.DAT or the STRECKEN.TOR on the other side.

2.2 The Links Data Table

The STRECKEN.DAT stores additional information for each link in the network that is required for running the Dortmund Model such as link category (six-lane motorway, four-lane motorway, rural road, major urban road, connector road, local road, pedestrians), traffic direction, type of change (no change, new construction, upgrade, deletion), the year in which the change took place or will take place, travel time and design speed (see Table 2.2).

Table 2.2 Attributes of the STRECKEN.DAT.

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Item Definition</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRART</td>
<td>2, 2, Integer</td>
<td>Link category code</td>
</tr>
<tr>
<td>RICHTUNG</td>
<td>1, 1, Integer</td>
<td>Driving direction code</td>
</tr>
<tr>
<td>MASART</td>
<td>2, 2, Integer</td>
<td>Change code</td>
</tr>
<tr>
<td>JAHR</td>
<td>5, 5, Integer</td>
<td>Year of change</td>
</tr>
<tr>
<td>VON</td>
<td>7, 7, Integer</td>
<td>From-node number</td>
</tr>
<tr>
<td>NACH</td>
<td>7, 7, Integer</td>
<td>To-node number</td>
</tr>
<tr>
<td>STRLANG</td>
<td>5, 5, Integer</td>
<td>Length of the link (in metres)</td>
</tr>
<tr>
<td>REISEZEIT</td>
<td>5, 5, Numeric, 1</td>
<td>Travel time (in minutes)</td>
</tr>
<tr>
<td>TEMPO</td>
<td>5, 5, Numeric, 1</td>
<td>Design speed (in km/h)</td>
</tr>
<tr>
<td>KNOTEN</td>
<td>14, 14, Integer</td>
<td>Redefined item representing both from- and to-node model numbers</td>
</tr>
</tbody>
</table>
Because the network is used to calculate changing inter-zonal travel times over time, past and future development and different temporal states of the network are recorded as pairs of attributes (year, kind of change) in this table. Subsequently, each link in the \textit{STRECKEN.AAT} may have more than one corresponding record in the \textit{STRECKEN.DAT} through a one-to-many relationship based on the unique identifier and the from- and to-nodes.

The link category code represents the different basic link categories as follows:

- 0 = Pedestrian walkway
- 1 = Motorway (6 lanes)
- 2 = Motorway (4 lanes)
- 3 = City motorway
- 4 = Main road (4 lanes)
- 5 = Main road (2 lanes)
- 6 = City road (2 lanes)
- 7 = Rural road (2 lanes)
- 9 = Access link
- 10 = Railway track
- 20 = S-Bahn track
- 30 = Underground track
- 40 = Tram track
- 50 = Bus line
- 60 = Underground + Bus / Tram + Bus / Underground + Tram

Figure 2.2 shows different link categories of the Dortmund region road network for the year 2003.

The driving direction code represents the driving direction as follows:

- 0 = Two-way street
- 1 = One-way street in from-to direction
- 2 = One-way street in to-from direction

The code of change represents the type of change which occurred to a link as follows:

- 0 = No change
- 1 = Deletion
- 2 = Change (category, direction, travel time, speed, etc.)
- 3 = New construction

The year of change indicates the year in which a change took place or will take place. Years are represented in this item as a 4 digit-number (e.g. 1998, 2002).

The from-node and to-node numbers accord with the Dortmund Model numbering system (see Node Attribute Table, Section 2.4). As a rule, the from-node number is always smaller than the to-node number.

The travel time signalises the travel time of public transport over this link in minutes.

The design speed is the allowed speed for cars in km/h.

The redefined item is composed of both the from-node and to-node numbers. It is unique for each link because each two nodes in the network are connected by one link only. This item is used as the key link between the \textit{STRECKEN.AAT} and the \textit{STRECKEN.DAT}. 
2.3 The Route Attribute Tables

The \texttt{STRECKEN.RAT<subclass>} stores information that describe public transport lines. Each transport line category (rail, urban and regional train (S-Bahn), tramway, underground and bus lines) is represented with an individual RAT table where all the information for the lines in this category are stored:

- \texttt{STRECKEN.RATEISENBAHN} for railway lines
- \texttt{STRECKEN.RATSBahn} for urban and regional train lines (S-Bahn)
- \texttt{STRECKEN.RATUBahn} for underground lines
- \texttt{STRECKEN.RATSTRASSENSBahn} for tram lines
- \texttt{STRECKEN.RATBUS} for bus lines

Each transport line is represented with one route which is in turn composed of several links comprising the path of the corresponding transport line. For each route in the network, information on line number, type of change, year of change, start-node, end-node and frequency of service are stored. Similar to the \texttt{STRECKEN.DAT}, past and future development and different
temporal states of the network are recorded for each individual transport line as pairs of attributes (year, kind of change) associated with an individual route (see Table 2.3).

Table 2.3 Attributes of the STRECKEN.RAT<subclass>.

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Item Definition</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINNUM</td>
<td>4, 4, Integer</td>
<td>Line number</td>
</tr>
<tr>
<td>MASART</td>
<td>2, 2, Integer</td>
<td>Change code</td>
</tr>
<tr>
<td>JAHR</td>
<td>5, 5, Integer</td>
<td>Year of change</td>
</tr>
<tr>
<td>VON1</td>
<td>7, 7, Integer</td>
<td>From-node number of section 1</td>
</tr>
<tr>
<td>NACH1</td>
<td>7, 7, Integer</td>
<td>To-node number of section 1</td>
</tr>
<tr>
<td>BHK1</td>
<td>7, 7, Integer</td>
<td>Service time intervals of section 1 (in minutes)</td>
</tr>
<tr>
<td>VON2</td>
<td>7, 7, Integer</td>
<td>From-node number of section 2</td>
</tr>
<tr>
<td>NACH2</td>
<td>7, 7, Integer</td>
<td>To-node number of section 2</td>
</tr>
<tr>
<td>BHK2</td>
<td>7, 7, Integer</td>
<td>Service time intervals of section 2 (in minutes)</td>
</tr>
<tr>
<td>VON3</td>
<td>7, 7, Integer</td>
<td>From-node number of section 3</td>
</tr>
<tr>
<td>NACH3</td>
<td>7, 7, Integer</td>
<td>To-node number of section 3</td>
</tr>
<tr>
<td>BHK3</td>
<td>7, 7, Integer</td>
<td>Service time intervals of section 3 (in minutes)</td>
</tr>
<tr>
<td>VON4</td>
<td>7, 7, Integer</td>
<td>From-node number of section 4</td>
</tr>
<tr>
<td>NACH4</td>
<td>7, 7, Integer</td>
<td>To-node number of section 4</td>
</tr>
<tr>
<td>BHK4</td>
<td>7, 7, Integer</td>
<td>Service time intervals of section 4 (in minutes)</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>3, 3, Integer</td>
<td>Symbol used to display the route</td>
</tr>
</tbody>
</table>

The line number is defined by a four-digit number (see Figure 2.3). The left most digit represent line type as follows:

1 = Rail
2 = Urban and regional train (S-Bahn)
3 = Underground
4 = Tram
5 = Bus

The second digit from left represents the public transport district to which the line belongs as follows:

1 = Unna
2 = Recklinghausen
3 = Bochum
4 = Dortmund
5 = Hagen
6 = Hamm
7 = Iserlohn

The last two digits represent the actual line number.
The change code represents the type of change which occurred to this line as follows:

- 0 = No change
- 1 = Deletion
- 2 = Change (path, time intervals, etc.)
- 3 = New construction

The year of change indicates the year in which a change took place or will take place. Years are represented in this item as a 4 digit-number (e.g. 1998, 2002).

Because the service time intervals may vary for different sections along a public transport line, the line can be divided into 4 sections. Each section is marked by a from-node and to-node number according to the Dortmund Model numbering system (see Node Attribute Table, Section 2.4). The four items BHK1….BHK4 are provided to define the service time intervals for each of those sections.

The symbol code is used by the Networks Module to display the route.

Figure 2.4 gives an overview of the public transport network of the Dortmund region for the year 2003.
2.4 The Node Attribute Table

The STRECKEN.NAT stores basic information that describe nodes in the network coverage. For each node of the network, information on geometrical location (Cartesian coordinates) and a unique identification number are stored in this file (see Table 2.4). The unique ID of each node is a six digit number representing the superzone (region) and the zone where the node is located.

Table 2.4 Attributes of the STRECKEN.NAT.

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Item Definition</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>HALT</td>
<td>30, 30, Character</td>
<td>Station or stop name</td>
</tr>
<tr>
<td>X-COORD</td>
<td>4, 12, Floating, 3</td>
<td>X-Coordinate</td>
</tr>
<tr>
<td>Y-COORD</td>
<td>4, 12, Floating, 3</td>
<td>Y-Coordinate</td>
</tr>
<tr>
<td>MODKNOT</td>
<td>7, 7, Integer</td>
<td>Node number</td>
</tr>
<tr>
<td>INTCODE</td>
<td>6, 6, Integer</td>
<td>The zone number where the node is located</td>
</tr>
</tbody>
</table>

The coordinates of the nodes accord with the Gauss-Krüger coordinate systems (Transverse Mercator projection).

The item MODKNOT contains the unique node number. The unique number of each node is a six digit number (see Figure 2.5) representing the superzone (most left pair of digits), the zone (middle pair of digits) where the node is located and the node number (most right pair of digits).

The item INTCODE comprises the zone number where the node is located. This is to be used by the Add Arc function of the Networks Module to assign node number to newly added nodes.

2.5 The Turn Off Restrictions Table

The STRECKEN.TOR contains information about turn off restrictions that is required for some transport models. Each link in the STRECKEN.AAT may have one corresponding record in the STRECKEN.TOR that reflects the restrictions at either of its end nodes. The information about turn off restrictions are driving direction-oriented. That means that there are two items that describe the turn off restrictions (see Table 2.5). The from-to-item describes the turn off restrictions at the to-node and the to-from-item describes the turn off restrictions at the from-node. Turn-over restrictions are not encoded in this table.

STRECKEN.TOR contains only those links that have turn off restrictions at one node at least. Links of the transport network without turn off restrictions are not represented in this table.
Table 2.5 Attributes of the STRECKEN.TOR.

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Item Definition</th>
<th>Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>VON</td>
<td>7, 7, Integer</td>
<td>From-node number</td>
</tr>
<tr>
<td>NACH</td>
<td>7, 7, Integer</td>
<td>To-node number</td>
</tr>
<tr>
<td>VN-TOR</td>
<td>1, 1, Integer</td>
<td>Turn off restriction in from-to direction</td>
</tr>
<tr>
<td>NV-TOR</td>
<td>1, 1, Integer</td>
<td>Turn off restriction in to-from direction</td>
</tr>
<tr>
<td>KNOTEN</td>
<td>14, 14, Integer</td>
<td>Redefined item representing both from- and to-node numbers</td>
</tr>
</tbody>
</table>

The from-node and to-node numbers accord with the Dortmund Model numbering system (see Node Attribute Table, Section 2.4). As a rule, the from-node number is always smaller than the to-node number.

The items VN-TOR and NV-TOR contain the direction oriented turn off restriction code as follows:

- 0 = No restriction
- 1 = Turning left is not allowed
- 2 = Turning right is not allowed
- 3 = Travelling ahead is not allowed
- 4 = Travelling is only allowed in the right direction
- 5 = Travelling is only allowed in the left direction

The redefined item is composed of both the from-node and to-node numbers. It is unique for each link because each two nodes in the network are connected by one link only. This item is used as the key link between the STRECKEN.AAT and the STRECKEN.TOR.
3. Export of Networks

The structure of the three ASCII datasets exported by the Networks Module to the Dortmund Model, the PROPOLIS-Raster Module and the ILUMASS Model differs tangibly due to the format accepted by each model. The structure of each of these output data is described in this chapter.

3.1 Export of Networks to the Dortmund Model

Transport networks are represented in the Dortmund Model as a unified multimodal network incorporating the modes road, public transport and bicycle/walk and indicating all changes occurring to a link between the base year and the simulation horizon, whereby a link may undergo more than one change. The Dortmund region transport networks is exported from ArchInfo by the Networks Module to the Dortmund Model in an ASCII file called \textit{ZVMxx.DAT} where xx is the scenario identifier. This file consists of two parts:

- \textit{Link Records} that provide information on each link on the network
- \textit{Line Records} that provide information on each transport line on the network

Both parts end with an end-statement of (99) that indicates the end of the part. The components and structure of each of these two parts can be described as follows:

3.1.1 Link Records of ZVMxx.DAT

The first part contains at least one record per link describing the attributes link category, type of change, year of construction, start-node, end-node, length in meters, travel time in minutes and design speed. The record format is as follows:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Length</th>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>integer</td>
<td>Public transport line code</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>integer</td>
<td>Link category</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>integer</td>
<td>Driving direction code</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>integer</td>
<td>Change code</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>7-10</td>
<td>4</td>
<td>integer</td>
<td>Year in which the change took place</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>12-17</td>
<td>6</td>
<td>integer</td>
<td>From-node number</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>19-24</td>
<td>6</td>
<td>integer</td>
<td>To-node number</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>26-30</td>
<td>5</td>
<td>integer</td>
<td>Length of the link (in metres)</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>32-35</td>
<td>4</td>
<td>real</td>
<td>Travel time (in minutes)</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>37-40</td>
<td>4</td>
<td>real</td>
<td>Design speed (in km/h)</td>
</tr>
</tbody>
</table>
The first three bytes compose the link code as follows:

The first byte of each record represents the type of the public transport line that travels over the link as follows:
- 0 = No line
- 1 = Railway track
- 2 = S-Bahn track
- 3 = Underground track
- 4 = Tram track
- 5 = Bus line
- 6 = Underground + Bus / Tram + Bus / Underground + Tram

The second byte of each record represents the type of the link as follows:
- 0 = Pedestrians / no street (railway, S-Bahn or underground track)
- 1 = Motorway (6 lanes)
- 2 = Motorway (4 lanes)
- 3 = City motorway
- 4 = Main road (3 lanes)
- 5 = Main road (2 lanes)
- 6 = City road (2 lanes)
- 7 = Rural road (2 lanes)
- 9 = Access link

The third byte of each record represents the direction of travel over this link as follows:
- 0 = Two-way street
- 1 = One-way street in from-to direction
- 2 = One-way street in to-from direction

The attribute change code represents the type of change that took place on this link as follows:
- 0 = No change
- 1 = Deletion
- 2 = Change (type, direction, travel time, speed etc.)
- 3 = New construction

If the link experiences one or more changes of one or more attributes or is removed during the simulation, the database contains multiple records for that link with time labels indicating the year in which the changes or the deletion took place. The year in which a change took place is specified at bytes seven to ten.

Figure 3.1 presents a small sample of the contents of this part of the final output file ZVMxx.DAT.

![Sample of the contents of part one of ZVMxx.DAT - link records.](image-url)
3.1.2 Line Records of ZVMxx.DAT

The second part of the file contains the public transport lines. Each public transport line is represented by a block of records. If the line experiences one or more changes of route or is removed during the simulation, the database contains multiple blocks of records for that link with time labels indicating the year in which the changes or the deletion took place.

The first record of each block serves as a header record, followed by a number of additional records storing the route alignment. For each line, line number, type of change, start-node, end-node and the frequency of service on different sections of the line are recorded in the header record. The record format is as follows:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Length</th>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>4</td>
<td>integer</td>
<td>Line number (see Section 2.3)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>integer</td>
<td>Change code</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>8-11</td>
<td>4</td>
<td>integer</td>
<td>Year in which the change took place</td>
</tr>
<tr>
<td>12-14</td>
<td>3</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>15-20</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 1</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>22-27</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 1</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>29-34</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 1</td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>36-41</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 2</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>43-48</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 2</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>50-55</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 2</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>57-62</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 3</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>64-69</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 3</td>
</tr>
<tr>
<td>70</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>71-76</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 3</td>
</tr>
<tr>
<td>77</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>78-83</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 4</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>85-90</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 4</td>
</tr>
<tr>
<td>91</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>92-97</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 4</td>
</tr>
</tbody>
</table>

The attribute change code represents the type of change that took place on this line as follows:

0 = No change
1 = Deletion
2 = Change (path, time intervals etc.)
3 = New construction

Because the service time intervals may vary for different sections along a public transport line (see Section 2.3), each section is marked by a from-node and to-node number. The node numbers accord with the Dortmund Model numbering system (see Node Attribute Table, Section
2.4). The service time intervals are given in minutes for up to four different sections (1–4) along the route of the transport line.

For the same transport line, every additional record presents the sequence of nodes traversed along the route of this line. The format of the records describing the route alignment of a public transport line is as follows:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Length</th>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>4</td>
<td>integer</td>
<td>Line number (see Section 2.3)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>integer</td>
<td>Change code</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>8-11</td>
<td>4</td>
<td>integer</td>
<td>Year in which the change took place</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>integer</td>
<td>Counter</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>15-20</td>
<td>6</td>
<td>integer</td>
<td>1st Node number</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>22-27</td>
<td>6</td>
<td>integer</td>
<td>2nd Node number</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>29-34</td>
<td>6</td>
<td>integer</td>
<td>3rd Node number</td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>36-41</td>
<td>6</td>
<td>integer</td>
<td>4th Node number</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>43-48</td>
<td>6</td>
<td>integer</td>
<td>5th Node number</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>50-55</td>
<td>6</td>
<td>integer</td>
<td>6th Node number</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>57-62</td>
<td>6</td>
<td>integer</td>
<td>7th Node number</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>64-69</td>
<td>6</td>
<td>integer</td>
<td>8th Node number</td>
</tr>
<tr>
<td>70</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>71-76</td>
<td>6</td>
<td>integer</td>
<td>9th Node number</td>
</tr>
<tr>
<td>77</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>78-83</td>
<td>6</td>
<td>integer</td>
<td>10th Node number</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>85-90</td>
<td>6</td>
<td>integer</td>
<td>11th Node number</td>
</tr>
<tr>
<td>91</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>92-97</td>
<td>6</td>
<td>integer</td>
<td>12th Node number</td>
</tr>
</tbody>
</table>

Byte thirteen of each record represents a counter for the statements describing the path of the transport line. A blank value indicates that this record is the header record in which the from-node, to-node and service time intervals for each line section are described. A sequence number is given under this column for each additional record that records the sequence of nodes traversed along the route of this transport line.

Figure 3.2 presents a small sample of the contents of this part of the final output file ZVMxx.DAT.
3.2 Export of Networks to the PROPOLIS-Raster Module

In contrast to the Dortmund Model described above, the PROPOLIS-Raster Module takes account of the geometry information of the network. The networks database is exported to the PROPOLIS-Raster Module using the Networks Module in a predefined ASCII data format. Four different ASCII files representing four sets of data are produced:

- Link Attributes File  NLDO.DAT
- Link Coordinates File  NLDO.LIN
- Node Attributes File  NNDO.DAT
- Node Coordinates File  NNDO.PTS

The contents and structure of these four files can be described as follows:

3.2.1 Link Attributes File (NLDO.DAT)

This file contains a set of attributes describing links that compose the transport network. For each link, a unique link number is associated with link type, start- and end-nodes, length and capacity. The record format is as follows:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Length</th>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>6</td>
<td>integer</td>
<td>Link-ID (unique link number)</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>8-13</td>
<td>6</td>
<td>integer</td>
<td>From-node number</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>15-20</td>
<td>6</td>
<td>integer</td>
<td>To-node number</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>integer</td>
<td>Link number if any</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>integer</td>
<td>PROPOLIS link code (see Table 3.1)</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>26-30</td>
<td>5</td>
<td>integer</td>
<td>Length of the link (in metres)</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>32-35</td>
<td>4</td>
<td>integer</td>
<td>Capacity of the link (in car/h)</td>
</tr>
</tbody>
</table>

Since the classification of link types in the PROPOLIS-Raster Module is standardised for all PROPOLIS case cities, it slightly differs from the classification used in the Dortmund Model and from the data stored in ArcInfo and hence needs some classification translation. Table 3.1...
shows the PROPOLIS-Raster module standard link type codes and their corresponding Dortmund Model codes.

Table 3.1 PROPOLIS-Raster module standard link type codes vs. Dortmund Model codes.

<table>
<thead>
<tr>
<th>PROPOLIS-Raster Module Code</th>
<th>PROPOLIS Link Type</th>
<th>Dortmund Model Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Motorway</td>
<td>1 = Motorway, six-lane</td>
</tr>
<tr>
<td>2</td>
<td>Motorway</td>
<td>2 = Motorway, four-lane</td>
</tr>
<tr>
<td>3</td>
<td>Motorway</td>
<td>3 = City motorway</td>
</tr>
<tr>
<td>2</td>
<td>Major urban road</td>
<td>4 = Main road, four-lane</td>
</tr>
<tr>
<td>5</td>
<td>Main road</td>
<td>5 = Main road, two-lane</td>
</tr>
<tr>
<td>3</td>
<td>Other roads</td>
<td>6 = City road, two-lane</td>
</tr>
<tr>
<td>7</td>
<td>Other roads</td>
<td>7 = Rural road, two-lane</td>
</tr>
<tr>
<td>4</td>
<td>Railway / Metro</td>
<td>10 = Rail</td>
</tr>
<tr>
<td>20</td>
<td>Urban and regional train (S-Bahn)</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Underground (U-Bahn)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Tramway and bus</td>
<td>40 = Tram</td>
</tr>
<tr>
<td>50</td>
<td>Bus</td>
<td>0 = Access link, intrazonal link by public transport / walking / cycling link</td>
</tr>
<tr>
<td>6</td>
<td>Access</td>
<td>9 = Access link, intrazonal link by car</td>
</tr>
</tbody>
</table>

The capacity of each link is calculated as a function of the number of lanes and link type according to the Dortmund Model link type codes. The value is given in cars per hour. Table 3.2 lists the different road capacities corresponding to the different link types.

Table 3.2 Capacity values according to link type.

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Road Capacity (Car/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2000</td>
</tr>
<tr>
<td>2</td>
<td>1800</td>
</tr>
<tr>
<td>3</td>
<td>1600</td>
</tr>
<tr>
<td>4</td>
<td>1200</td>
</tr>
<tr>
<td>5</td>
<td>700</td>
</tr>
<tr>
<td>6</td>
<td>400</td>
</tr>
<tr>
<td>7</td>
<td>450</td>
</tr>
</tbody>
</table>

Figure 3.3 on the next page presents a small sample of the contents of the final output file NLDO.DAT.
3.2.2 Link Coordinates File (NLDO.LIN)

This file contains sets of pairs of coordinates describing the exact alignment of each link. This geometry part of the data is produced using the ArcInfo command UNGENERATE and is formatted according to the ArcInfo GENERATE data format. This is an ASCII format composed of a unique number for each link followed by pairs of coordinates describing the vertices along the link. The key word END separates the information of the different links. Link unique number is the same as the link-id described in the attribute section above and used to establish the relation between the attributes section and the geometry section. All coordinates are in the Gauss-Krüger coordinate system (Transverse Mercator projection). Figure 3.4 presents a small sample of the produced links geometry file.

```
1 2601638.2500000 5701206.0000000
2601665.7500000 5701086.5000000
2601713.7500000 5700965.5000000
2601737.2500000 5700918.0000000
2601849.0000000 5700939.5000000
2601947.0000000 5700961.0000000
2601953.5000000 5700962.0000000
2602052.2500000 5701001.0000000
2602117.7500000 5700653.5000000
END

2 2601638.2500000 5701206.0000000
2601599.2500000 5701171.0000000
... END END
```

Figure 3.4 Sample of the contents of the links geometry file (NLDO.LIN).

3.2.3 Node Attributes File (NNDO.DAT)

This file contains two attributes that describe each node. These are the node identification number (Node-ID) and node model number (see Figure 3.5). The node model number (Node Number) is the same number listed under From and To in the NLDO.DAT file (see Figure 3.3, Section 3.2.1).
3.2.4 Node Coordinates File (NNDO.PTS)

This file contains a set of pairs of coordinates describing the exact location of each node along with the node unique number. This geometry part of the data is produced using the ArcInfo command **UNLOAD** which is applied on the **STRECKEN.NAT** (see Section 2.4). The node identification number (Node-ID) is the key link between the attributes file (NNDO.DAT) and the coordinates file (NNDO.PTS). All coordinates are in the Gauss-Krüger coordinate system (Transverse Mercator projection). Figure 3.6 presents a small sample of the produced nodes geometry file.

![Node-ID and Coordinates](image)

**Figure 3.6 Sample of the contents of the nodes geometry file (NNDO.PTS).**

3.3 Export of Networks to the ILUMASS Model

The exchange of network data between the GIS database and the ILUMASS Model is implemented through two ASCII-Files:

- **Links and Transport Lines Data File**  
  TR_xx_yyyy.DAT

- **Links Geometry File**  
  TR_xx_yyyy.LIN

where xx is the scenario number, and yyyy is the reference year.
Both files are produced by the Networks Module in the format accepted by the ILUMASS Model. The links and transport lines data file \textit{TR\_xx\_yyyy.DAT} is composed of two different parts which follow each other. The first part represents the records that describe link attributes (see Section 3.3.1), where the second part represents the records that describe the transport lines attributes (see Section 3.3.2). Both parts end with an end-statement of (99) that indicates the end of the part.

The links geometry file \textit{TR\_xx\_yyyy.LIN} contains sets of pairs of coordinates describing the exact alignment of each link (see Section 3.3.3).

The format and codes of each of these two files are described below.

### 3.3.1 Link Records of \textit{TR\_xx\_yyyy.DAT}

A link in the network is the line between two nodes. Only one link exists between each two nodes of the network. Unlike the link records part of \textit{ZVMxx.DAT} file (see Section 3.1.1), this first part of \textit{TR\_xx\_yyyy.DAT} contains only one record per link. This record represents the most up-to-date status of the link at the reference year (yyyy). The record format is as follows:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Length</th>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>6</td>
<td>integer</td>
<td>From-node number</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>8-13</td>
<td>6</td>
<td>integer</td>
<td>To-node number</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>integer</td>
<td>Public transport line code</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>integer</td>
<td>Link category</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>integer</td>
<td>Driving direction code</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>19-23</td>
<td>5</td>
<td>integer</td>
<td>Length of the link (in metres)</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>25-28</td>
<td>4</td>
<td>real</td>
<td>Travel time (in minutes)</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>30-33</td>
<td>4</td>
<td>real</td>
<td>Design speed (in km/h)</td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>integer</td>
<td>Intersection code (from-to direction)</td>
</tr>
<tr>
<td>36</td>
<td>1</td>
<td>integer</td>
<td>Turning code (from-to direction)</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>38</td>
<td>1</td>
<td>integer</td>
<td>Intersection code (to-from direction)</td>
</tr>
<tr>
<td>39</td>
<td>1</td>
<td>integer</td>
<td>Turning code (to-from direction)</td>
</tr>
</tbody>
</table>

The three attributes public transport line code, link category and driving direction code compose the link code as follows:

The public transport line code represents the type of the public transport line that travels over the link as follows:

- 0 = No line
- 1 = Railway track
- 2 = S-Bahn track
- 3 = Underground track
- 4 = Tram track
- 5 = Bus line
- 6 = Underground + Bus / Tram + Bus / Underground + Tram
The link category represents the type of the link as follows:

0 = Pedestrians / no street (railway, S-Bahn or underground track)
1 = Motorway (6 lanes)
2 = Motorway (4 lanes)
3 = City motorway
4 = Main road (3 lanes)
5 = Main road (2 lanes)
6 = City road (2 lanes)
7 = Rural road (2 lanes)
9 = Access link

The driving direction code represents the direction of travel over this link as follows:

0 = Two-way street
1 = One-way street in from-to direction
2 = One-way street in to-from direction

The intersection code represents the direction-oriented type of intersection as follows:

0 = Motorway, travel priority
1 = Traffic light
2 = Stop / priority to traffic from right
3 = No intersection / Bus stop / Underground or Tram station

The turning code represents the direction-oriented rule controlling turns as follows:

0 = No restriction
1 = Turning left is not allowed
2 = Turning right is not allowed
3 = Travelling ahead is not allowed
4 = Travelling is only allowed in the right direction
5 = Travelling is only allowed in the left direction

Due to the fact that the ILUMASS Model requires information on the types of road intersections, and since these information are not directly given by the database of the transport network of the Dortmund region (see Chapter 2), these information is generated automatically within the Networks Module based on simple rules.

For the automatic coding, nodes where at least three links (streets) intersect, are to be considered as an intersection. Nodes with one or two intersecting links can be seen as end-nodes or public transport stops.

To determine the road intersection type the following three rules are adopted:

1) If only one or two links meet at a node, the intersection code is set to 3 (no road intersection).

2) If more than two links intersect at a node and if there are more than two different link types of the intersecting links, then the road intersection is assumed to be controlled by a traffic light. The intersection code is then set to 1 (traffic light controlled intersection).

3) If more than two links with less than three different link types intersect at a node, then the type of road intersection depends on the two intersecting link types. The available intersection codes are shown in the matrix in Figure 3.7.
<table>
<thead>
<tr>
<th>From</th>
<th>Motorway (cp ≈ 2000 Cars/h)</th>
<th>City motorway (cp ≈ 1600 Cars/h)</th>
<th>Main road (four-lane, cp ≈ 1200 Cars/h)</th>
<th>Main road (two-lane, cp ≈ 700 Cars/h)</th>
<th>City road (two-lane, cp ≈ 400 Cars/h)</th>
<th>Rural road (two-lane, cp ≈ 450 Cars/h)</th>
<th>Access link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motorway</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>City motorway</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Main road (four-lane)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Main road (two-lane)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>City road (two-lane)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Rural road (two-lane)</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Access link</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

cp = roadway capacity

Road intersection types:
0 = Motorway, travel priority
1 = Traffic light
2 = Stop / priority to traffic from right
3 = No intersection / bus stop / underground or tram station

Figure 3.7  Matrix for determining types of road intersections.

Figure 3.8 presents a small sample of the contents of this part of the final output file TR_xx_yyy.DAT.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Code</th>
<th>Length</th>
<th>Speed</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>110301</td>
<td>110303</td>
<td>050</td>
<td>795</td>
<td>0.5</td>
<td>70</td>
</tr>
<tr>
<td>110301</td>
<td>110304</td>
<td>640</td>
<td>1204</td>
<td>1.2</td>
<td>35</td>
</tr>
<tr>
<td>110301</td>
<td>110306</td>
<td>450</td>
<td>555</td>
<td>1.0</td>
<td>35</td>
</tr>
<tr>
<td>110302</td>
<td>110306</td>
<td>440</td>
<td>230</td>
<td>0.4</td>
<td>35</td>
</tr>
<tr>
<td>110302</td>
<td>110307</td>
<td>050</td>
<td>519</td>
<td>0.9</td>
<td>35</td>
</tr>
<tr>
<td>110303</td>
<td>110308</td>
<td>300</td>
<td>890</td>
<td>1.5</td>
<td>35</td>
</tr>
</tbody>
</table>

...
3.3.2 Line Records of TR_xx_yyyy.DAT

The second part of this file contains the public transport lines of the reference year (yyyy). Each public transport line is represented by a block of records. The first record of each block serves as a header record, followed by a number of additional records storing the route alignment. Line number, type of change, start-node, end-node and the frequency of service on different sections of the line are recorded in the header record. The format of the header record is as follows:

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Length</th>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>4</td>
<td>integer</td>
<td>Line number (see Section 2.3)</td>
</tr>
<tr>
<td>5-7</td>
<td>3</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>8-13</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 1</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>15-20</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 1</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>22-27</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 1</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>29-34</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 2</td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>36-41</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 2</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>43-48</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 2</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>50-55</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 3</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>57-62</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 3</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>64-69</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 3</td>
</tr>
<tr>
<td>70</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>71-76</td>
<td>6</td>
<td>integer</td>
<td>From-node number of section 4</td>
</tr>
<tr>
<td>77</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>78-83</td>
<td>6</td>
<td>integer</td>
<td>To-node number of section 4</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>85-90</td>
<td>6</td>
<td>integer</td>
<td>Service time interval of section 4</td>
</tr>
</tbody>
</table>

Because the service time intervals may vary for different sections along a public transport line (see Section 2.3), each section is marked by a from-node and to-node number. The node numbers accord with the Dortmund Model numbering system (see Node Attribute Table, Section 2.4). The service time intervals are given in minutes for up to four different sections (1–4) along the route of the transport line.

For the same public transport line, every additional record presents the sequence of nodes traversed along the route of this line. The format of the records describing the route alignment of a public transport line is as follows:
<table>
<thead>
<tr>
<th>Bytes</th>
<th>Length</th>
<th>Type</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>4</td>
<td>integer</td>
<td>Line number (see Section 2.3)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>integer</td>
<td>Counter</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>8-13</td>
<td>6</td>
<td>integer</td>
<td>1st Node number</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>15-20</td>
<td>6</td>
<td>integer</td>
<td>2nd Node number</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>22-27</td>
<td>6</td>
<td>integer</td>
<td>3rd Node number</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>29-34</td>
<td>6</td>
<td>integer</td>
<td>4th Node number</td>
</tr>
<tr>
<td>35</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>36-41</td>
<td>6</td>
<td>integer</td>
<td>5th Node number</td>
</tr>
<tr>
<td>42</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>43-48</td>
<td>6</td>
<td>integer</td>
<td>6th Node number</td>
</tr>
<tr>
<td>49</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>50-55</td>
<td>6</td>
<td>integer</td>
<td>7th Node number</td>
</tr>
<tr>
<td>56</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>57-62</td>
<td>6</td>
<td>integer</td>
<td>8th Node number</td>
</tr>
<tr>
<td>63</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>64-69</td>
<td>6</td>
<td>integer</td>
<td>9th Node number</td>
</tr>
<tr>
<td>70</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>71-76</td>
<td>6</td>
<td>integer</td>
<td>10th Node number</td>
</tr>
<tr>
<td>77</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>78-83</td>
<td>6</td>
<td>integer</td>
<td>11th Node number</td>
</tr>
<tr>
<td>84</td>
<td>1</td>
<td>blank</td>
<td>-</td>
</tr>
<tr>
<td>85-90</td>
<td>4</td>
<td>integer</td>
<td>12th Node number</td>
</tr>
</tbody>
</table>

The sixth byte of each record represents a counter for the statements describing the path of the public transport line. A blank value indicates that this record is the header record in which the from-node, to-node and service time intervals for each line section are described. A sequence number is given under this column for each additional record that records the sequence of nodes traversed along the route of this transport line.

Figure 3.9 presents a small sample of the contents of this part of the final output file.

![Image of a table](IRPUD)

**Figure 3.9** Sample of the contents of part two of TR_xx_yyyy.DAT – public transport line records.
3.3.3 Links Geometry File (TR_xx_yyyy.LIN)

This file contains sets of pairs of coordinates describing the exact alignment of each link. This geometry part of the data is produced using the ArcInfo command `UNGENERATE` and is formatted according to the ArcInfo `GENERATE` data format. This is an ASCII format composed of a unique number for each link followed by pairs of coordinates describing the vertices along the link. The key word END separates the information of the different links. The link unique number is the combination of the from- and to-node numbers described in the attribute section above and used to establish the relation between the attributes file and the geometry file. All coordinates are in the Gauss-Krüger coordinate system (Transverse Mercator projection). Figure 3.10 presents a small sample of the produced link geometry file.

![Diagram of Link-IDs and Coordinates](image)

**Figure 3.10** Sample of the contents of the links geometry file (TR_xx_yyyy.LIN).
4. The Networks Module Interface – User Guide

In this chapter the interface of the Networks Module is described. It is intended for the users who may need to modify the transport networks data. It explains the use and functions of the graphical user interface (GUI) of the Networks Module and its components.

The GUI of the Networks Module is composed of four main components: the main window, the Display submodule, the Edit submodule and the Output submodule. The function of each component and the tools offered by each submodule can be described as follows:

4.1 The Main Window

The main window of the GUI of the Networks Module is subdivided into four main elements: the Main Display Screen, the Zoom/Pan Tools, the Main Menu and the Attributes List Table (see Figure 4.1).

The Main Display Screen displays basically the transport networks of the Dortmund region for editing along with any other optional background data such as the administrative layer, the cartographic map or the land use layer of the region. All features are symbolised and displayed on this screen according to user commands. The display allows the interaction for zooming in/out or roaming the display according to the tool selected from the Zoom/Pan tools as well as for the selection of features from the current edit feature mode.

The Zoom/Pan Tools allow the user to zoom in or out, pan or refresh the display. Full view display or zoom to selected function are also offered. The user needs to select one of these
tools, moves the mouse cursor to the main display, and applies the tool on the current display. Figure 4.2 presents the function of each of these tools.

**Figure 4.2 Functions of the different Zoom/Pan tools.**

- **Zoom to the full extent button:** Creates a full view display.
- **Zoom in tool:** Zooms in to a certain point. User should press 9 to end this function.
- **Zoom out tool:** Zooms out to a certain point. User should press 9 to end this function.
- **Zoom in window tool:** Zooms in to a certain box defined by the user.
- **Pan tool:** Pans the display by dragging it in any direction. User should press 9 to end this function.
- **Zoom to selected features button:** Zooms the display to the currently selected features.
- **Refresh button:** Redraws the display.

*The Main Menu* comprises the core menu of the module interface. It contains three different submenus, each of which represents one of the three submodules (i.e. Display, Edit and Output submodules) and has a set of tools that implement the function of this submodule. When the user selects a certain submodule by clicking the corresponding button at the top of this menu, the menu changes to reflect the tools associated with this submodule. A detailed description of these three submodules will follow in this chapter.

*The Attributes List Table* is designed to list and update attributes associated with the features in the network. It is composed of scrollable lists with columns representing ArcInfo items, while the rows represent the values under these items. The values listed in this list can be updated by direct interaction as it will be explained later under the Edit submodule.

### 4.2 The Display Submodule

The Display submodule provides the necessary tools to display and symbolise the features of the network database (see Figure 4.3). The Display submenu which contains the display tools is activated by selecting the red **Display** button on top of the main menu.

The **Year** input box enables the user to specify a certain year of study. Only street links and transport lines established before or completed in this year are shown. The years supported by the database range between 1950 and 2003 for already existing features and 2004 and 2030 for planned features. This function is designed to help the user to display and edit features within a focused representation of the database limited to the features available before a specific year. That can be seen as zooming the database in and out through the time.
The Public Transport section allows the user to show and/or hide any combination of the public transport lines in predefined symbols. The same applies for the Streets section that lists the different street categories available to enable the user to choose any combination of them. Clicking on the Draw button confirms the user’s settings and displays the selected street categories. Similarly, the Nodes section offers the possibility to show the nodes with/without identification number or to hide the nodes.

The Background combo-box offers a set of options comprising several possible layers that can be displayed in the background to guide the user during the editing process. Offered background layers are the land use layer and the administrative layer of the Dortmund region. Moreover, a cartographic raster map can be displayed as well.

The Display submodule can be seen as a helpful orientation tool that guides the users through the spatial as well as the temporal dimension of the database. The user can easily switch between the Edit and the Display submodules. It is for example very useful to show node numbers while public transport lines are being modified or to refer to the land use layer while a new street is being digitised.

Figure 4.3 The Display submodule.
4.3 The Edit Submodule

The Edit submodule provides the user with the necessary tools to edit, modify and update the transport network features and their associated attributes (see Figure 4.4). In this context, it can be seen as the core submodule of the Networks Module. The Edit submenu is activated by selecting the green **Edit** button on top of the main menu.

![Figure 4.4 The Edit submodule.](image)

4.3.1 Setting Initial Parameters

Before using any commands or functions, the type of feature to be edited has to be specified. In the **Editing Settings** Box one can choose between arcs, nodes and routes. The default edit feature class is arc. When the user chooses to edit routes, a combo box is activated for selecting the special public transport class. Here, the user can choose between bus, tram, underground (U-Bahn), urban and regional railway (S-Bahn) and rail.

An **Edit Distance** and a **Node Snap** distance can be specified by the user. The **Intersect Arcs** option can be set to on or off. The edit distance is a search tolerance used for feature selection. A feature must be within the edit distance circle to become selected. The node snap distance is a search tolerance used for node snapping. When a new feature is added, the Networks Module will search the entire area inside the node snap distance from the node under consid-
eration and snap to the closest node found. The default edit distance is set to $1/100^\text{th}$ and the default node snap distance is set to $1/1000^\text{th}$ of the width or height of the STRECKEN coverage, whichever is greater. The user can change the distances either by directly writing the new distance as a positive integer or real number in the correspondent text box or by clicking with the mouse on the correspondent button which is located over the text box. Then the distance can be specified by digitising two points at the main display screen. The Intersect Arcs option provides the possibility to split arcs automatically and to add nodes when arcs are added. If the Intersect Arcs option is off, only the node snap function proceeds but no intersections will be calculated. This is the default setting.

4.3.2 The Selection Tools

The Select tools allow the selection of one or many feature(s) of the activated feature class in different ways as shown in Figure 4.5.

- Select many: selects one or many feature(s) from the current feature class by clicking on the features to be selected until 9 is pressed.
- Select by box: selects all features within a specific box defined by the user. Features do not have to be completely contained within the selection area to be selected.
- Select by attributes: selects feature(s) based on a certain attribute (e.g. from- and to-node number, line number etc.).
- Clear selection: removes all selected features from the currently selected set.

*Figure 4.5 The selection tools.*

Selected features are coloured with a magenta colour. The number of selected features is incessantly displayed in the text box No. Selected for a better overview.

The attributes of the selected feature(s) can be listed by clicking on the List Data button of the Edit menu. The attributes associated with the selected feature(s) are shown in the Attributes List Table that is located in the lower area of the main form (see Figure 4.7). Listing the attributes of the selected feature(s) clears the selection set. The user can easily select a record from the list for editing and updating attributes.

4.3.3 Updating Attributes

To alter the attributes of features, records have to be selected in the Attributes List Table. Attributes of both links and transport lines are listed in the attribute lists (see Figure 4.7) by clicking the List Data button (see Figure 4.6). Individual records can then be updated by highlighting any value in the table and changing it. As described before, each state of a link is stored as a record in the STRECKEN.DAT (see Section 2.2). If a certain link experiences a change, new records can be added or modified using the attribute editing tools (see Figure 4.6).
Lists all records of the selected feature(s) in the attribute lists.

Adds a new record to the selected link in the \texttt{STRECKEN.DAT}.

Deletes the selected record from the attribute lists and from \texttt{STRECKEN.DAT}.

\textit{Figure 4.6 The attributes editing tools.}

The \textbf{Add Record} function updates the \texttt{STRECKEN.DAT} attribute table by creating a new record for a selected link, populating each item in the table with the appropriate attributes for this link, updating the start-node and end-node items for the link and automatically calculating and updating the new travel time based on the actual length and the designed speed. The user has to determine new link code and year of change.

The \textbf{Delete Record} function deletes the selected record. This function should be used with caution. While features are not saved until the \texttt{Save} button is pressed, all updated attributes or added and deleted records are immediately saved to the \texttt{STRECKEN.DAT} table. Deleted records cannot be restored. The \texttt{UNDO} function applies only for edits in the spatial data.

Clicking the id-number of a certain record select this record for editing. One or many feature(s) can be selected at a time. Selected records are highlighted in a blue background colour. To edit a certain attribute value, the user has to double click either on the item name or on the value to be edited. This will open an input box directly above the item to be modified (see Figure 4.7). New values are to be entered in this input box. Pressing carriage return will update the highlighted value with the entered value.

\textit{Figure 4.7 The Attributes List Table and attribute modification.}

Not all listed attributes are changeable by the user. System internal variables cannot be changed. For feature class arc, only link code, change code, year, travel time and speed can be updated and for nodes only station name can be changed. Attributes of route features can all be changed.

\subsection*{4.3.4 Editing Links and Public Transport Lines}

The Networks Module makes available a set of tools that allow the user to add or modify road network links or public transport lines. It utilises ArcInfo’s editing capabilities and provides some add-ons to facilitate the addition or alteration of features. Links and routes can be added, copied or deleted using the editing tools presented in Figure 4.8.
Adds a new link or a new public transport line to the STRECKEN coverage. For adding a new public transport line all corresponding arcs must be selected beforehand.

Copies a selected public transport line. The new public transport line has the same extent like the original. User specific attributes have to be entered manually.

Deletes a link or a whole public transport line.

Figure 4.8 The feature editing tools.

The Networks Module helps editing road intersections, bus stops and rail stations. For each added or modified feature, the six-digit unique node identification number (UID) is automatically updated. The tool searches for newly added nodes (with UID = 0). Then it reads the zone ID from the zone where each new node is located, identifies the node UIDs that already exist within this zone and assigns a new number for the new node so that it is unique in the zone. Finally, it assigns a new UID for this node composed of the zone ID and the node number so that the node UID is unique in the whole network. Link attributes are then updated by transferring newly added node UIDs from the NAT to the from-node and to-node items of the AAT. At last, the user has to enter user specific attributes supported by an attributes input form (see Figure 4.9).

Figure 4.9 The attributes input form for a new link (l.) or a new public transport line (r.).

In addition to the functions mentioned above, two additional tools (see Figure 4.10) are offered for editing routes that represent public transport lines. Routes can be extended, shortened or altered while the corresponding RAT is updated accordingly. Therefore the user has to select respective links with the mouse.
Shortens the currently selected public transport route by removing sections.

Extends the currently selected public transport route by adding sections.

**Figure 4.10 The routes editing tools.**

### 4.3 The Output Submodule

The main task of this submodule (see Figure 4.11) is to produce the final output files in the format accepted by the Dortmund Model, the PROPOLIS-Raster Module and the ILUMASS Model. The Output submenu is activated by selecting the yellow **Output** button on top of the main menu.

The Dortmund Model section allows the user to select the layers to be exported to the Dortmund Model. The user can choose to export the network, the land use and/or the administrative layer. Selecting the **Network** check box extracts relevant link attributes from the network.
representation in ArcInfo and converts them to the equivalent representation of the first part (links records) of the Dortmund Model network representation. It also converts the ArcInfo route representation of public transport lines to the equivalent representation of the second part (lines records) of the Dortmund Model network representation (see Section 3.1). Finally, it combines the information of the road and public transport network and incorporates the information about network changes into one multimodal network scenario file with multiple records for each link or line which undergoes changes. At the end of this process the main display screen is turned into a large scrollable window (see Figure 4.11) listing the produced ASCII file for visual review. Selecting the Land use or the Administrative units check box exports the underlying land use or administrative units layer into the format accepted by the Dortmund Model.

The Link attributes check box of the PROPOLIS-Raster Module section extracts the up-to-date relevant roads and public lines attributes along with node numbers from the network representation in ArcInfo and converts them to the format needed by the PROPOLIS-Raster Module as described earlier (see Section 3.2). The Network geometry check box produces an additional file that contains the coordinates of nodes defining the exact path of each road in the network. Again, in all cases the resulted ASCII file is displayed in the main display screen.

The two ASCII-files needed by the ILUMASS Model can be produced by indicating a Reference year (yyyy) and a Scenario number (nn). The appropriate check boxes should be activated for generating the required file in the format described earlier (see Section 3.3). Network Attributes are stored in a file called TR_nn_yyyy.DAT whilst Network Geometry is stored in a file named TR_nn_yyyy.LIN.
5. Technical Description – Developer Guide

This Chapter is intended for programmers who may wish to advance and modify the current version of the Networks Module (Version 1.0) in the future. Before all technical functions of the system are elaborated, the technical specifications, the auxiliary input data and the Networks Module system structure are briefly described.

5.1 Technical Specifications

5.1.1 Software

The Networks Module is built to run on ArcInfo 8.x Workstation. VB version 5.0 or higher is needed to modify the system but is not required to run it for a normal use since it is delivered and run as an executable file.

5.1.2 Hardware

The Networks Module is designed to work on a Windows-based computer with the following specifications as a minimum:

- 700 MHz processor speed
- 64 MB of RAM
- 100 MB storage space

5.1.3 Input Data

The input data to the Networks Module can be distinguished under two different data types:

Main input data

These are the data sets that contain the links, lines and nodes of the transport network together with their associated attributes and are edited directly by the user. Such data sets were described in details under Chapter 2 “Representations of Networks in ArcInfo” and can be listed here as follows:

- **STRECKEN** coverage that contains the road links and public transport lines of the transport network of the Dortmund region.
- **STRECKEN.AAT** info table that contains link basic attributes and model Ids.
- **STRECKEN.DAT** info table that contains the time series records of links attributes.
- **STRECKEN.RAT<SubClass>** that contains the attributes of the transport lines under the different subclasses as follows:
  - **STRECKEN.RATEISENBAHN** for railway lines
  - **STRECKEN.RATUBAHN** for urban and regional train lines (S-Bahn)
  - **STRECKEN.RATUBAHN** for underground lines
  - **STRECKEN.RATSTRASSENBAHN** for tram lines
  - **STRECKEN.RATBUS** for bus lines
- **STRECKEN.NAT** info table that contains nodes attributes.
- **STRECKEN.TOR** info table that contains turn of restrictions at intersections.

The standard input data format is the current version of ArcInfo Coverage. The co-ordinate system used in this data is the Gauss-Krüger coordinate system (Transverse Mercator projection).

**Auxiliary input data**

These are the data sets that are used to support and facilitate the implementation of the Networks Module during the run time. They are not accessed directly by the user, but they are very important for the system developer. These data sets are described separately in detail in Section 5.2 “Auxiliary Input Data”.

5.1.4 Output Data

With the Networks Module two different types of outputs can be produced:

- ArcInfo coverage: an edited version of the **STRECKEN** coverage and its related tables.
- ASCII files: the three different groups of ASCII files for the Dortmund Model, PROPOLIS-Raster Module and the ILUMASS Model (see Chapter 3 “Export of Networks”):
  - Dortmund Model: ZVMxx.DAT
  - PROPOLIS-Raster Module: NLDO.DAT, NLDO.LIN, NNDO.DAT and NNDO.PTS
  - ILUMASS Model: TR_xx_yyyy.DAT and TR_xx_yyyy.LIN

5.1.5 Distribution

The Networks Module is distributed to the end users as an executable file (NetworksModule.exe) along with a set of AML Macros, ArcInfo symbol sets, icons and ArcInfo coverages. The Module is also available as a Visual Basic Project (NetworksModule.vbp) for further modifications by the developers. The components of the working folder are shown in Figure 5.1 and can be described as follows:

The system is contained in the **NetworksModule** folder where all the data, symbol sets, temporary files and output files are located. It contains the following items:

- **AMLS** is a folder that contains the AML macros that are run and accessed by the system.
- **Dotifs** is a folder that contains the scanned images of Dortmund used for the background display in a TIFF format.
- **info** is the ArcInfo system folder that contains the attribute tables.
- **landuse** is a polygon coverage representing the land use of the Dortmund region used for the background display.
- **modbez1** is a polygon coverage representing the zoning system of the Dortmund region for the background display as well as for numbering nodes.
Figure 5.1 The contents of the Networks Module folder.

- **strecken** is a line coverage representing the street links as well as public transport lines of the Dortmund region transport networks. Historical links attributes are recorded over the time and are stored in an external info table (**STRECKEN.DAT**).

- **strecken-lst** is a copy of the **STRECKEN** coverage representing the latest status of each link and transport line in the network stored in the **STRECKEN-LTST.AAT** table. This coverage is produced by the system when the user attempt to export the data to the ILUMASS model.

- **strecken-ras** is a copy of the **STRECKEN** coverage representing the latest status of each link and transport line in the network stored in the **STRECKEN-RAS.AAT** table. Links ids are modified in this coverage so that a serial number is assigned. This coverage is produced by the system when the user attempts to export the data to the PROPOLIS-Raster module.
- **VB-Modules** is a folder that contains all the VB modules and forms used by the system. This folder is not required for running the system from the executable file, but it is required for system modification.

- **IRPUD.lin** is a line symbol set created at IRPUD and is used to symbolize arcs and routes.

- **IRPUD.shd** is a shade symbol set created at IRPUD and is used to symbolize polygons of the LANDUSE and the MODBEZ1 coverages.

- **na.exe** is an executable Fortran program written at IRPUD. This software helps to identify potential errors in the ZVMxx.DAT file. The output is the ZVMERR file that contains the errors found in the data.

- **NetworksModule.exe** this is the executable file of the Networks Module that should be run by the user.

- **NetworksModule.vbp** this is the Visual Basic project that should be opened when the system is to be modified or updated.

- **Waiting.avi** is the animation file which is called by frmWait that indicates that the system is processing a certain task.

### 5.2 Auxiliary Input Data

The auxiliary data sets are used to support and facilitate the implementation of the Networks Module during the run time. They are needed as a part of a certain task during the execution of the Networks Module. They may be needed to improve the display (e.g. background data and symbol sets). They can also be used to establish certain relations among the different info tables (e.g. relation files). Another use is to act as a template for certain temporary info tables that may be established during the run time of the system. Some ASCII files are also used temporarily to support a certain task within the system.

These data sets are not accessed or modified by the user, but they are important for the execution of the system. Missing some of these data during the run time of the system, may result in an error and the user will be warned through a warning message. In some cases, the system may even fail to continue if the missing data is essential for its execution.

These data sets can be described as follows:

#### 5.2.1 Coverages

- **LANDUSE** coverage that is used as a background display to improve user’s orientation during the editing process. It is a polygon coverage where polygons are classified according to the different land use categories stored under an item called *LEM* in LANDUSE.PAT info table.

- **MODBEZ1** coverage that contains the zoning system of the region. It is mainly used to calculate the model-id number of new nodes according to their location within one of these zones. It can also be displayed in the background to facilitate the editing process.
It is a polygon coverage where each polygon represents a certain zone. Zone numbers are stored under an item called INTCODE in MODBEZ1.PAT.

5.2.2 Info Tables

- **LINIENATTRIBUTES.ORG** is an empty info table that is used as a template for the route attribute table of the different route subclasses in the STRECKEN coverage. It has the item definitions of a typical route attribute table without any records. It is used by a set of AMLs to restore the route system in case it is distorted after an editing session.

- **MAXJAHR.STS** is a result of performing the Statistics function of ArcInfo on the STRECKEN.DAT table to find the most up-to-date record for each link. MAXJAHR.STS describes for each link the latest year when the link was changed. It is a temporary file that is produced and used by the system while exporting the data to the PROPOLIS-Raster Module and the ILUMASS Model.

- **RASTER** is an ArcInfo image catalog that stores the path to the TIFF images of the Dortmund region in an item called IMAGE that points to the images stored under the DoTifs folder. This raster catalog is used by the system to display the images in the background as a means of orientation.

- **STRECKEN.REL** is a relation info file that stores two relations used by the system to establish the relation between STRECKEN.AAT and STRECKEN.DAT. Those two relations are AAT2DAT, which relates both tables based on the KNOTEN item, and LNK2MOD, which relates the same tables based on the MODEL-ID item. STRECKEN.REL is restored upon loading the Networks Module. Both relations are used throughout the whole system.

- **STRECKEN<yyyy>.DAT** is a temporary copy of the STRECKEN.DAT info table that stores the records for the links that are built up to a reference year (yyyy). This is used to restrict the display to show only the links available at the time point (yyyy) when the user enters this year in the Year input-text in the Display submodule. The file STRECKEN2030.DAT is the default file that shows all links. These tables are created and used by the system.

5.2.3 Symbol Sets

- **IRPUD.LIN** is the line symbol set used to symbolise links and routes.

- **IRPUD.SHD** is the shade symbol set used to symbolise polygons.

5.2.4 ASCII Files

- **ShowData.txt** is an ASCII temporary file that is used as an interface between ArcInfo info tables and Visual Basic controls to store the attributes of links or routes. Attributes of selected features are read from the corresponding attribute table in ArcInfo and are written out in a certain format to ShowData.txt. Then it is read by the Visual Basic routines, which are responsible for attribute listing, to show the attributes within the attrib-
ute lists. Again, as a temporary file, ShowData.txt is created, restored and maintained by the system.

5.3 System Structure

The Networks Module is based on the Open Development Environment (ODE) approach which allows the development of powerful and visually appealing ARC/INFO applications with any of the development environments that support ActiveX controls such as Visual Basic, Delphi, or Visual C++ (ESRI, 2002). Existing AML scripts can still be incorporated into applications developed within an ODE. This approach was utilized in building the Networks Module using two main programming elements:

- **Arc Macro Language (AML):** AML is the script language of ArcInfo Workstation. It represents the core code of the Networks Module where all the main tasks and function are performed in ArcInfo. AML code is written in the so called AML Macros which are ASCII files with the extension (aml).

- **Visual Basic (v. 5.0):** Visual Basic is mainly used in this application to allow the development of professional and user-friendly interface that hides the complexity of the underlying code. In addition, it allows the compilation of the application in an executable form that can be distributed among the different computers as long as ArcInfo license is available on those computers. Visual Basic code is written in a set of modules that compose the main Visual Basic Project (NetworksModule.vbp) and are distributed in an executable file format (NetworksModule.exe).

The Networks Module is designed in a way that allows the user to edit the STRECKEN coverage, which represents the transport network of the Dortmund region, and to produce the required output ASCII files for the different models described earlier. To do so the system is structured in a way that offers a user-friendly interface that hides the complexity of the underlying processes. The graphic user interface is composed of a set of User-Forms and VB-Modules, which in turn call the relevant ArcInfo functions either directly or through a set of AMLs. The STRECKEN coverage is then edited, queried and exported by using the suitable ArcInfo and/or AML functions. Figure 5.2 illustrates this system structure.

![Figure 5.2 The Networks Module system structure.](image)
5.4 Technical Description

The Networks Module is composed of four main components: the Main-Form, the Display submodule, the Edit submodule and the Output submodule. The functionality and use of each of these components are described in Chapter 4 “The Networks Module Interface – User Guide”. The focus in this section is on the technical description of each of these components and their sub-components. The following pages describe systems components, the relation and the data flow among these components and the external entities that deal with the system. The terminology and standards of the Structured Systems Analysis and Design Method-SSADM (Weaver, 1993) is used to describe the software components and to design the data flow diagrams.

5.4.1 SSADM Terminology

A set of symbols and icons are going to be used in the following pages to facilitate and improve the understanding of the system description. These symbols and icons and their meanings are described below in Figure 5.3.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process or task that works inside the system. This can be a function, an event or an AML macro.</td>
<td></td>
</tr>
<tr>
<td>External entity that deals with the system by sending or receiving data or commands.</td>
<td></td>
</tr>
<tr>
<td>Digital database where data sets are stored and accessed.</td>
<td></td>
</tr>
<tr>
<td>Command flow from one entity or process to the other.</td>
<td></td>
</tr>
<tr>
<td>One way data flow between two different processes or between one process and a database.</td>
<td></td>
</tr>
<tr>
<td>Two way data flow between two different processes or between one process and a database.</td>
<td></td>
</tr>
<tr>
<td>Visual Basic event.</td>
<td></td>
</tr>
<tr>
<td>Visual Basic module or sub procedure.</td>
<td></td>
</tr>
<tr>
<td>Visual Basic form or control on a form.</td>
<td></td>
</tr>
<tr>
<td>AML macro.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.3 The SSADM terminology and symbols used in the system description.

5.4.2 Overall System Description

The Networks Module is composed of four main submodules and several functions. Figure 5.4 is a context diagram that illustrates the overall system components and the external entities dealing with the system.
This context diagram, shows the overall system of the Networks Module and its major components as a single process with data flow between the system and the outside relevant elements represented by external entities. This context diagram sets up the boundaries of the system. To develop this context diagram, the sources and recipients of data from the system as external entities, together with the major data flows to and from these external entities, were identified in Table 5.1.

Table 5.1 All sources (S) and recipients (R) and the data flow between the Networks Module and the external entities.

<table>
<thead>
<tr>
<th>External Entity</th>
<th>Source/Recipient</th>
<th>Data Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>S</td>
<td>Original STRIEKEN coverage and info tables</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>Other coverages for the background display</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>Edited STRIEKEN coverage and info tables</td>
</tr>
<tr>
<td>Symbol sets</td>
<td>S</td>
<td>ArcInfo symbol sets: IRPUD.shd, IRPUD.lin</td>
</tr>
<tr>
<td>The Dortmund Model</td>
<td>R</td>
<td>ZVMxx.DAT</td>
</tr>
<tr>
<td>The PROPOLIS-Raster Module</td>
<td>R</td>
<td>NLDO.DAT, NLDO.LIN, NNDO.DAT, NNDO.PTS</td>
</tr>
<tr>
<td>The ILUMASS Model</td>
<td>R</td>
<td>TR_xx_yyyy.DAT, TR_xx_yyyy.LIN</td>
</tr>
</tbody>
</table>

From the context diagram and Table 5.1, the system life cycle can be described as follows:

- The Main Form loads the STRIEKEN coverage from ArcInfo GIS database.
- It also calls the symbol sets and assigns initial symbols to the different feature classes.
- The Display Submodule is used to load and show/hide layers in the fore and background of the display. Layers are always called from the database.
- The Edit submodule is used to edit and modify the STRIEKEN coverage and the info tables and to save it back to the database.
- Finally, the Export Submodule exports the STRECKEN coverage to the different external models and modules.

5.4.3 Detailed System Description

Each of the four systems components are to be described here in more detail. This description is focused on the **Purpose** of each component and the **Functions** which are implemented within each component. It gives an overall description of each function, its, **Inputs** and **Outputs**. Limitations, planned enhancements and any comments to the described function are outlined as **Notes**. A simple **Data Flow Diagram** (DFD) is presented for each function to illustrate its components in the source code (forms, modules, routines, events, AMLs) and the relationships among them. It also shows the relationship between the different internal components of each function and the external entity they deal with. The functionality of each component is described briefly within the same diagram. If one of the functions is called by certain controls in the interface, this control will be pointed out in a small snapshot showing its position in the **Interface**.

General purpose and support AMLs, forms and modules are described at the end of this section under “Auxiliary and Support Tools”.
Main Form (0.1)

**Purpose:** The *Main Form* is the main interface that the users interact with. It contains the main ArcEdit1 display canvas and all the VB controls such as lists, buttons, check-boxes and combo-boxes.

**Functions:** Four main functions are implemented within the *Main Form*:

0.1.1 Set initial parameters
0.1.2 React to resizing
0.1.3 Call sub-modules
0.1.4 Bail out

**Inputs:** The STRECKEN Coverage and its associated attribute tables:

- STRECKEN.AAT
- STRECKEN.NAT
- STRECKEN.DAT
- STRECKEN.REL

The symbol sets: IRPUD.lin and IRPUD.shd

**Outputs:** None

![Figure 5.5 The DFD of the “Main Form”](image_url)

---

**Figure 5.5 The DFD of the “Main Form”**
Set initial parameters (0.1.1)

Purpose: To declare and initiate the variables and parameters required during the runtime life of the Networks Modules.

Inputs: The STRECKEN coverage and its associated attribute tables:
- STRECKEN.AAT
- STRECKEN.NAT
- STRECKEN.DAT
- STRECKEN.RAT<subclass>
- STRECKEN.REL

The symbol sets: IRPUD.lin and IRPUD.shd

Outputs: Initial size parameters: current width, height and screen resolution

Figure 5.6 The DFD of the “Set initial parameters” function.
React to resizing (0.1.2)

**Purpose:** This function helps to maintain the look of the form if its size has been changed or if the screen resolution has been changed. It adjusts the size of all controls within the form according to the new size of the form if the form size will be changed by the user. It also changes the font size of control names in order to fit the new size.

**Inputs:** Size parameters: current width, height and screen resolution

**Outputs:** None

![Diagram](image)

Figure 5.7 The DFD of the “React to resizing” function.
Call sub-modules (0.1.3)

**Purpose:** To call the relevant sub-module (i.e. Display, Edit or Output Modules) according to the chosen button and to hide the other modules.

**Inputs:** None

**Outputs:** None

<table>
<thead>
<tr>
<th>0.1.3</th>
<th>Call sub-modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1.3.1</td>
<td>frmMain::cmdDisplay</td>
</tr>
<tr>
<td>- Hide the Edit frame (frmEdit)</td>
<td></td>
</tr>
<tr>
<td>- Hide the Output frame (frmOutput)</td>
<td></td>
</tr>
<tr>
<td>- Show the Display frame (frmDisplay)</td>
<td></td>
</tr>
<tr>
<td>0.1.3.2</td>
<td>frmMain::cmdEdit</td>
</tr>
<tr>
<td>- Hide the Display frame (frmDisplay)</td>
<td></td>
</tr>
<tr>
<td>- Hide the Output frame (frmOutput)</td>
<td></td>
</tr>
<tr>
<td>- Show the Edit frame (frmEdit)</td>
<td></td>
</tr>
<tr>
<td>0.1.3.3</td>
<td>frmMain::cmdOutput</td>
</tr>
<tr>
<td>- Hide the Display frame (frmDisplay)</td>
<td></td>
</tr>
<tr>
<td>- Hide the Edit frame (frmEdit)</td>
<td></td>
</tr>
<tr>
<td>- Show the Output frame (frmOutput)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.8 The DFD of the “Call sub-modules” function (left) and the relevant buttons on the interface (right).

Bail out (0.1.4)

**Purpose:** To check if the edit changes have been saved before quitting the Networks Module. If the changes are not saved, this function warns the user and asks if the changes should be saved before bailing out.

**Inputs:** None

**Outputs:** None

<table>
<thead>
<tr>
<th>0.1.4</th>
<th>Bail out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1.4.1</td>
<td>frmMain::UnLoad</td>
</tr>
<tr>
<td>- Search for edit changes</td>
<td></td>
</tr>
<tr>
<td>- Check if changes should be saved</td>
<td></td>
</tr>
<tr>
<td>- If yes, save changes and quit</td>
<td></td>
</tr>
<tr>
<td>- If no, quit without saving</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.9 The DFD of the “Bail out” function.
Display Module (0.2)

**Purpose:** The Display Module is responsible for the display and symbolisation of data within the Networks Module. In addition, it provides the users with the tools to zoom in/out, pan and refresh the display. The module also allows the display of different background layers as means of orientation to the users.

**Functions:** Six main functions are implemented within the Display Module:
- 0.2.1 Zoom/Pan Tools
- 0.2.2 Display time snapshots
- 0.2.3 Display public transport
- 0.2.4 Display street categories
- 0.2.5 Display nodes
- 0.2.6 Display background data

**Inputs:** Background datasets:
- The land-use coverage LANDUSE
- The zones coverage MODBEZ1
- The image catalog of the Dortmund region DoTifs

**Outputs:** None

![Diagram of the Display Module (0.2)](image)

*Figure 5.10 The DFD of the “Display Module”.*
Zoom/Pan tools (0.2.1)

*Purpose:* To provide the user with the necessary tools to zoom in/out, pan and refresh the display. It also allows drawing the display to its full extent and zooming to a certain selected feature.

*Inputs:* None

*Outputs:* None

---

```
0.2.1 Zoom/Pan tools
  0.2.1.1 cmdFull_Click
  - Draw the display to its full extent
  0.2.1.2 cmdRefresh_Click
  - Redraw the display
  0.2.1.3 cmdZoomIn_Click
  - Zoom in the display according to one point click
  0.2.1.4 cmdZoomOut_Click
  - Zoom out the display according to one point click
  0.2.1.5 cmdZoomExtent_Click
  - Zoom in the display according to a certain extent
  0.2.1.6 cmdPan_Click
  - Pan the display dynamically
  0.2.1.7 cmdZoomSelect_Click
  - Zoom to the selected set of features
```

---

*Figure 5.11* The DFD of the “Zoom/Pan Tools” and the relevant buttons on the interface.
Display time snapshots (0.2.2)

**Purpose:** To restrict the displayed features to a certain time snapshot. That means to display only features that were built up to a certain reference year.

**Inputs:** A reference year.

**Outputs:** None

---

Figure 5.12  The DFD of the “Display time snapshots” function and the relevant text-box on the interface.
Display public transport lines (0.2.3)

Purpose: To symbolise, show or hide selected public transport lines.

Inputs: None

Outputs: None

Display links (0.2.4)

Purpose: To show or hide the selected link categories.

Inputs: None

Outputs: None

Figure 5.13 The DFD of the “Display routes” function (left) and the relevant check-boxes on the interface(right).

Figure 5.14 The DFD of the “Display links” function and the relevant controls on the interface.
Display nodes (0.2.5)

Purpose: To show or hide nodes and their model identification number

Inputs: None

Outputs: None

Figure 5.15 The DFD of the “Display nodes” function (left) and the relevant option-buttons on the interface (right).

Display background data (0.2.6)

Purpose: To show/hide different data sets in the background

Inputs: None

Outputs: None

Figure 5.16 The DFD of the “Display background data” function (left) and the relevant combo-box on the interface (right).
Edit Module (0.3)

**Purpose:** The *Edit Module* offers the necessary tools to edit, modify and update the network features and their associated attributes. In this context, it can be seen as the core scenario builder module where the transport network is to be changed and modified to generate the different scenarios. Editing transactions are implemented mainly within the ArcEdit object (*ArcEdit1*) in the form of commands sent from the interface to the ArcEdit object.

**Functions:** Seven main functions are implemented within the *Edit Module*:
- 0.3.1 Set editing settings
- 0.3.2 Selection tools
- 0.3.3 Edit attributes
- 0.3.4 Edit arcs
- 0.3.5 Edit routes
- 0.3.6 Edit nodes
- 0.3.7 Undo and Save edit changes

**Inputs:** The *STRECKEN* coverage and its associated attribute tables:
- *STRECKEN.AAT*
- *STRECKEN.NAT*
- *STRECKEN.DAT*
- *STRECKEN.REL*

The editing settings:
- edit distance
- node snap
- intersect arcs status
- edit features

**Outputs:** None
Figure 5.17 The DFD of the “Edit Module”.
Set editing settings (0.3.1)

**Purpose:** To set initial editing settings and tolerances such as edit feature, edit distance, node snap and intersect arcs.

**Inputs:**
- Default edit distance of the edit coverage **STRECKEN**
- Default node snap of the edit coverage **STRECKEN**

**Outputs:**
- Edit feature (arc, route.<subclass> or node)
- Edit distance
- Node snap distance
- Intersect arcs status (add or off)

**Figure 5.18** The DFD of the “Set editing settings” function.

** Figure 5.19** The interface of the “Set editing settings” function.
Selection tools (0.3.2)

**Purpose:** To allow the user to select features through different means.

**Inputs:** None

**Outputs:** Selected set of features

**Notes:** Select by attribute (frmSelAtt) allows the user to select features based on a certain attribute. While arcs can be selected either by model-id or from and to node numbers, routes can only be selected by line number and nodes can be selected by node number.

---

**Figure 5.20** The DFD of the “Selection tools” (left) and the relevant controls on the interface (right).
Edit attributes (0.3.3)

**Purpose:** To allow the user to list and edit the attributes associated with selected features.

**Inputs:** Selected set of features

**Outputs:** None

![Figure 5.21 The Buttons of the “Edit attributes” function on the interface (left) and the attribute list (right).]
Figure 5.22 The DFD of the “Edit attributes” function.
Edit arcs (0.3.4)

Purpose: To add and modify arcs and their associated attributes.

Inputs: None

Outputs: None

Figure 5.23 The DFD of the “Edit arcs” function.

Figure 5.24 The Buttons of the “Edit arcs” function on the interface (left) and the “New Arc Attributes” form (right).
Edit routes (0.3.5)

**Purpose:** To add and modify routes and their associated attributes.

**Inputs:** None

**Outputs:** None

---

**Figure 5.25** The DFD of the “Edit routes” function and the relevant buttons on the interface.
Edit nodes (0.3.6)

The Edit node function should allow the user to add or delete nodes and to modify node attributes. The current version of the Networks Module supports the modification of node attributes through the attribute lists - frmMain.List1(). The addition and deletion of nodes is not supported in this version. Refer to Chapter 6 “Future Developments and Enhancements” for more details on the recommended tools under this function.

Save/Undo edit changes (0.3.7)

**Purpose:** To provide the user with the tools to save the edit changes to the STRECKEN coverage and to undo any edit transaction that has been performed since the last save.

**Inputs:** None

**Outputs:** None

**Notes:** The Save button saves the edit changes to the STRECKEN coverage. In the current version, scenarios that may be built by the user in the future are all saved to the same coverage.

The Undo button undoes the most recent change to the features of the STRECKEN coverage but it does not undo the changes that may have been performed to the link attributes in STRECKEN.DAT. This limitation is because the STRECKEN.DAT table is an external table, and changes in external info tables is saved automatically to the original version of the table. One solution to undo adding records is to delete added records by using the Delete Record button (see Edit Attributes). Another workaround in the current version is to make a copy of STRECKEN.DAT to STRECKEN.ORG at the load event of the frmMain. Like that, users can go back to STRECKEN.ORG every time the original attributes of the links are desired.

**Figure 5.26 The DFD of the “Save/Undo edit changes” function (left) and the relevant Buttons on the interface (right).**
Output Module (0.4)

**Purpose:** The Output Module is intended to allow the user to export the GIS-based transport networks to the required output format accepted by the Dortmund Model, the ILUMASS Model, or the PROPOLIS-Raster Module. These three output formats are described earlier in Chapter 4 “Export of Networks”.

**Functions:** Three main functions are implemented within the Output Module:
- 0.4.1 Export to the Dortmund Model
- 0.4.2 Export to the PROPOLIS-Raster Module
- 0.4.3 Export to the ILUMASS Model

**Inputs:** The STRECKEN coverage and its associated attribute tables:
- STRECKEN.AAT
- STRECKEN.NAT
- STRECKEN.DAT
- STRECKEN.REL
- STRECKEN.TOR

**Outputs:**
- ZVMxx.DAT to the Dortmund Model
- NLDO.DAT, NLDO.LIN, NNDO.DAT and NNDO.PTS to the PROPOLIS-Raster Module
- TR_xx_yyyy.DAT and TR_xx_yyyy.LIN to the ILUMASS Model

![Figure 5.27 The DFD of the “Output Module”](image)

---

**Figure 5.27 The DFD of the “Output Module”**
Export to the Dortmund Model (0.4.1)

**Purpose:** To export the transport links and public transport lines to the ZVMxx.DAT ASCII file accepted by the Dortmund Model.

**Inputs:** The **STRECKEN** coverage and its associated attribute tables:
- STRECKEN.AAT
- STRECKEN.NAT
- STRECKEN.DAT
- STRECKEN.REL

**Outputs:** ZVMxx.DAT, where xx is the scenario number.

**Notes:** The export of land-use data is not integrated in this version of the Networks Module. The Land use check box is offered here for future versions in which it is planned to integrate the export of the land-use data to this button. Refer to Chapter 6 “Future Developments and Enhancements” for more information.

---

**Figure 5.28** The DFD of the “Export to the Dortmund Model” function and the relevant controls on the interface.
Export to the PROPOLIS-Raster Module (0.4.2)

**Purpose:** To export the transport links and nodes attributes and geometry to the PROPOLIS-Raster Module.

**Inputs:** The STRECKEN coverage and its associated attribute tables:
- STRECKEN.AAT
- STRECKEN.NAT
- STRECKEN.DAT
- STRECKEN.REL

**Outputs:**
- Network links attributes: NLDO.DAT
- Network links geometry: NLDO.LIN
- Network nodes attributes: NNDO.DAT
- Networks nodes geometry: NNDO.PTS

**Notes:** The activation of the output files is intended to allow the user to select which data to export. Since the four ASCII files were always generated together, this function to export one or more of them separately has not yet been integrated. This may be useful for future development of the system. Otherwise, if not needed, this function and its associated check-boxes can be removed.

---

**Figure 5.29** The DFD of the “Export to the PROPOLIS-Raster Module” function and the relevant controls on the interface.
Export to the ILUMASS Model (0.4.3)

Purpose: To export the transport links and public transport lines to the TR_xx_yyyy.DAT and TR_xx_yyyy.LIN ASCII files accepted by the ILUMASS Model.

Inputs: The STRECKEN coverage and its associated attribute tables:
- STRECKEN.AAT
- STRECKEN.NAT
- STRECKEN.DAT
- STRECKEN.REL
- STRECKEN.TOR

Outputs: - Transport attributes data: TR_xx_yyyy.DAT
- Transport geometry: TR_xx_yyyy.LIN

where xx is the scenario number and yyyy is the reference year

Notes: The current version of AMLs written to export the networks to the ILUMASS Model does not support the reference year and scenario number. Both output files are produced up to the reference year 2000 and the basic scenario 00. However, the possibility of specifying the reference year and scenario number are offered here to encourage the integration of both in future versions of the Networks Module. Again, refer to Chapter 6 for more details.
Figure 5.30 The DFD of the “Export to the ILUMASS Model” function and the relevant controls on the interface.
Auxiliary and Support Tools (0.5)

Purpose: The forms, modules and AMLs described here can be seen as auxiliary tools that support the four main submodules of the Networks Module and can be called by any of them. In the same time they do not belong to any of the sub-modules and cannot be classified as part of any of them. Thus, they are gathered here under this title and described in detail.

Functions: Seven tools are considered here as auxiliary tools:

- 0.5.1 frmIntro
- 0.5.2 Database
- 0.5.3 Clear and set symbol item
- 0.5.4 Disable and enable all controls
- 0.5.5 lbl9Message
- 0.5.6 frmWait
- 0.5.7 Route restoration scheme

Inputs: Differ depending on the tool

Outputs: Differ depending on the tool
frmIntro (0.5.1)

**Purpose:** This is the splash screen that appears at the beginning when the user runs the Networks Module. *frmIntro* is displayed while the Networks Module is being opened and the initial parameters are being set, showing a progress bar that indicates the percentage of progress.

**Inputs:** None

**Outputs:** None

**Notes:** The adjustment of the speed in which the progress bar is progressed is to be set during design time by the programmer. It depends on the speed of the machine on which the Networks Module is to be run. The time the module requires to open should be estimated and tested for each machine. Then this time should be divided by 10 to calculate how many seconds, or part of a second, the module needs while opening. This time intervals in seconds should then be adjusted in the timer *Timer1*.

![DFD Diagram](image)

**Figure 5.31** The DFD of the “frmIntro” Form (left) and the splash screen (right).

Database (0.5.2)

**Purpose:** The term *Database* here means the GIS database that stores the Dortmund region networks in ArcInfo.

**Inputs:** None

**Outputs:** None
Clear and set symbol item (0.5.3)

**Purpose:** The Networks Module uses an item called *Symbol* in the Route Attribute Table as a symbol item to symbolise the different routes that represent the transport lines. It was found that some processes are run much slower after assigning such a symbol item in ArcEdit than before this assignment. The tool *mdlDraw.ClearSymbolItem* is used to clear the symbol items before running certain processes, like the drawing process, in order to speed up such processes. The symbol items are then reset back after the process is over by using the *mdlDraw.SetSymbolItem*

**Inputs:** None

**Outputs:** None

---

![Figure 5.32 The DFD of the “Clear and set symbol item” function.](image)
Disable and enable all controls (0.5.4)

Purpose: These two routines `mdlDraw.DisableAll` and `mdlDraw.EnableAll` are used to disable/enable all the controls in the `frmMain`. Disabling all the controls before running a certain process prevents that the user clicks a certain control while the process is still in progress and hence prevents sending two commands to ArcEdit in the same time. Sending a command to ArcInfo while it is processing another command can cause unexpected results, including fatal errors (ESRI, 2002).

Inputs: None

Outputs: None

![Figure 5.33 The DFD of the “Disable and enable all controls” function.](image)

Lbl9Message (0.5.5)

Purpose: This is a simple label control that is used to display the message “Press 9 to finish”. It is called with the Visible property set to True before performing tasks that wait that the user presses 9 to finish. Once the user presses 9, the label is called with the Visible property set to False. Otherwise, it has no events or associated code.

Inputs: None

Outputs: None
**frmWait (0.5.6)**

*Purpose:* This form is displayed while a certain process is running to indicate that the system is running and that the user should wait.

*Inputs:* None

*Outputs:* None

![Diagram of frmWait](image)

*Figure 5.34 The DFD of the “frmWait” Form (left) and the message (right).*
Route restoration scheme (0.5.7)

**Purpose:** Editing routes in ArcEdit has a common bug that results in a total or partial distortion of the route system. In most cases, this only affects the lastly edited route subclass. To overcome this problem, the so called *Route Restoration Scheme* was developed for the Networks Module to allow the restoration of the distorted route subclass(es). It simply reads the affected subclass from an ASCII file called *Linien.txt* and rebuilds the whole route system. *Linien.txt* contains the transport lines part of the latest error-free *ZVMxx.DAT* data. In such case the user loses only the last changes in the route system since this *ZVMxx.DAT* was produced rather than loosing the whole route subclass.

**Notes:** This function has not been integrated in the Networks Module interface. It is to be called from ArcInfo command line. It is planned to integrate this function in future versions (see Chapter 6 “Future Developments and Enhancements”).

**Inputs:**
- *Linien.txt:* an ASCII file that contains the transport lines part of *ZVMxx.DAT*
- STRECKEN coverage

**Outputs:** New route system in the STRECKEN coverage

![Figure 5.35 The DFD of the “Route restoration scheme” function.](image)

<table>
<thead>
<tr>
<th>0.5.7</th>
<th>Route restoration scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5.7.1</td>
<td>CreateRoutesFromLinienTxt.AML</td>
</tr>
<tr>
<td>- Run CreateStopsFileFromLinienTxt.Aml to create stops and Attribute files</td>
<td></td>
</tr>
<tr>
<td>- Load resulted file Stops.txt into Strecken.stp</td>
<td></td>
</tr>
<tr>
<td>- Load LinienAttributes.dat into an info file</td>
<td></td>
</tr>
<tr>
<td>- Build the selected route subclass(es)</td>
<td></td>
</tr>
<tr>
<td>- Join the LinienAttributes.dat to the relevant route attribute table in the strecken coverage</td>
<td></td>
</tr>
<tr>
<td>- Delete all temporary files.</td>
<td></td>
</tr>
<tr>
<td>0.5.7.2</td>
<td>CreateStopsFileFromLinienTxt.AML</td>
</tr>
<tr>
<td>- Open and read routes Linien.txt</td>
<td></td>
</tr>
<tr>
<td>- Open output files Stops.txt &amp; LinienAttribute.txt for writing</td>
<td></td>
</tr>
<tr>
<td>- Read each line of Linien.txt</td>
<td></td>
</tr>
<tr>
<td>- Compose the attributes of each line and write them to LinienAttribute.txt</td>
<td></td>
</tr>
<tr>
<td>- Compose the nodes of each line and write them to Stops.txt</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 5.35 The DFD of the “Route restoration scheme” function.**
6. Future Developments and Enhancements

The current version of the Networks Module (Version 1.0) is fully operational and provides the necessary tools for displaying, editing, updating and exporting the Dortmund region transport network. The available tools support most of the required operations and transactions that may be carried out by the user. However, a set of enhancements is suggested for the future versions to increase the efficiency and to further ease the usage of the interface. These enhancements can be classified as follows:

6.1 Enhancements to the Display Submodule

6.1.1 Different Symbolization to Different Link Categories

One of the important enhancements to the Display Submodule is to display the different link categories in different colours and symbols as it is the case for the public transport lines. A unique line symbol for each street type should enhance the clarity of the display. Such symbols should reflect the category of links by setting the right thickness and colour. The selection colour should then be different to the various link category symbols. A symbol item may be added to the arc attribute table `STRECKEN.AAT`, where symbol codes for each arc can be assigned and retrieved.

6.1.2 Map Composition and Printing Current Screen

The current version of the Networks Module lacks a function that allows the user to compose and print maps of the whole or part of the transport network. Layouts and maps are still to be produced outside the system (e.g. in ArcPlot or ArcView 3.x). The integration of such function within the Display Submodule or the Output Submodule is a useful addition to the system. Simple maps that present the transport network at a certain reference year can support the editing process by helping the user comparing such maps to the source maps. In addition, maps that show the development of the street network over the time, or the differences in the public transport lines between two points of time in a certain area can be very useful for transport and regional planners. Another simple function can be developed to allow the user to print the current display.

6.2 Enhancements to the Edit Submodule

6.2.1 Adding Node/Splitting Link

A new tool that allows the addition of new nodes or splitting an existing arc into two new arcs by inserting a node is very important in the editing process. Such a tool should allow the user to simulate the case in which an existing street is divided into two or more portions. The new portions of the street should automatically inherit all the attributes of the original street. The user should then be asked to enter a new record of attributes that indicates this division. This tool can benefit from the developed sub-routine `frmMain.AddNode` in assigning a new model
node number to the added node. In the same context, deleting an existing node or unsplitting an existing link should also be offered as an additional tool.

6.2.2 Attribute Names Instead of Attribute Codes

Another editing enhancement is to replace attribute codes, which are listed in the attribute lists, by their textual values or names. This will allow new users, who may not be familiar with the attribute codes, to work with the Networks Module without the need to learn the different codes. Working with common self-explanatory attribute names should definitely ease and facilitate the editing process specially for new users.

6.2.3 Selecting Nodes by Station Name

So far nodes can be selected only by node number. It was noticed that in many cases, users may need to select a certain station and to zoom in to this station. Selecting nodes by the station name attribute is another function that is suggested for future development of the system. This requires not only the integration of the necessary code to perform this function, but also the completion of the station name attribute (Halt) in the node attribute table STRECKEN.NAT.

6.2.4 Selecting Routes by End Stations

In the same context like for selecting nodes by station name, a new function may be added to allow selecting route by from- and to-nodes. This will allow the user to select all transport lines that travel between two specific stations. Users should be able to define the end stations either by node number or by station name. The possibility to select all public transport lines that start or end at a certain station is another useful addition.

6.2.5 Integrating the Routes Restoration Scheme

As mentioned under Section 5.4.3 “Detailed System Description”, a route restoration scheme is developed to rebuild the route system in case it is distorted or lost after editing. This scheme is run in the moment out side the Networks Module and can only be called from ArcInfo command line. This scheme can be integrated within the Networks Module interface in future versions.

6.2.6 Scenario Manager

One of the main purposes of the Networks Module is to generate different scenarios of the Dortmund region transport network by carrying out some changes in the network. When this point is reached, it is very important to come up with the appropriate method to save together the different changes of the network that belong to a certain scenario. This can be seen as a scenario manager that stores and retrieves the features relevant to each scenario under a unique scenario name. A straightforward approach is to save the changes of each scenario under a new coverage named to the scenario name. But this solution consumes a lot of disk
space because of the redundancy that results from saving the coverage several times with the little changes that belong to each scenario. A better solution is to assign a certain attribute to the changes that belong to a certain scenario so that only those changes can be pre-selected before exporting the network under this scenario. This attribute can be added to the feature attribute tables and to the links data table STRECKEN.DAT to mark records according to their corresponding scenarios.

6.3 Enhancements to the Output Submodule

6.3.1 Error Detection Schemes

Several error checking procedures are developed and integrated within the Networks Module in order to validate the data and attributes entered by the user. However, a set of error detection schemes can still be developed and integrated within the system to help the user identifying and correcting possible errors. One example is a node model number checker, which checks that no node number has a zero value and that all node numbers are unique in the whole network. It may also check that all nodes that fall within a certain zone, have this zone number as a part of their node model number (see Section 2.4). A link error checker may be another example that checks that each link has a unique model number and that each link has the necessary basic attributes such as the link code, the allowed speed and the travel time.

One important step in this direction is to integrate the NA.EXE software which checks the consistency of the ZVMxx.DAT and hence the whole data. This software detects and lists the errors and conflicts in the data in an external ASCII file called ZVMERR. The NA.EXE software should be integrated into the interface so that the data can be checked and the errors can be identified before the ZVMxx.DAT file is delivered to the Dortmund Model.

6.3.2 Exporting Land-Use Data

The development of the necessary tools to export the land-use data to the Dortmund Model is also seen as an important addition to the Output Submodule. Only then, the Networks Module can be seen as the primary tool for editing and exporting data required by the supported transport models. This will make the users to deal with one interface to export all required data.

6.3.3 Reference Year and Scenario Number for Export to ILUMASS

The current version of the Networks Module exports the networks to the ILUMASS Model for one reference year (2000) and for the basic scenario (00). The production of the output files for different reference years and the building of different scenarios is an essential function that should be integrated in the Output Submodule in the future. For that reason, the possibility of specifying the reference year and scenario number are already offered in the interface and need only to be coded.
6.4 General Enhancements

Based on previous experience and the experience gained while developing the Networks Module, it was found that procedures written in VB run faster than scripts that perform the same function written in AML. Therefore it is highly recommended, whenever possible, to replace the AML code with the equivalent VB code that performs the same functions. This should enhance the performance of the Networks Module tangibly.
References


