GIMA

Geographical Information Management and Applications

Towards a 3D geo-data model to support pedestrian routing in multimodal public transport travel advices.

MSc Thesis

23 June 2010

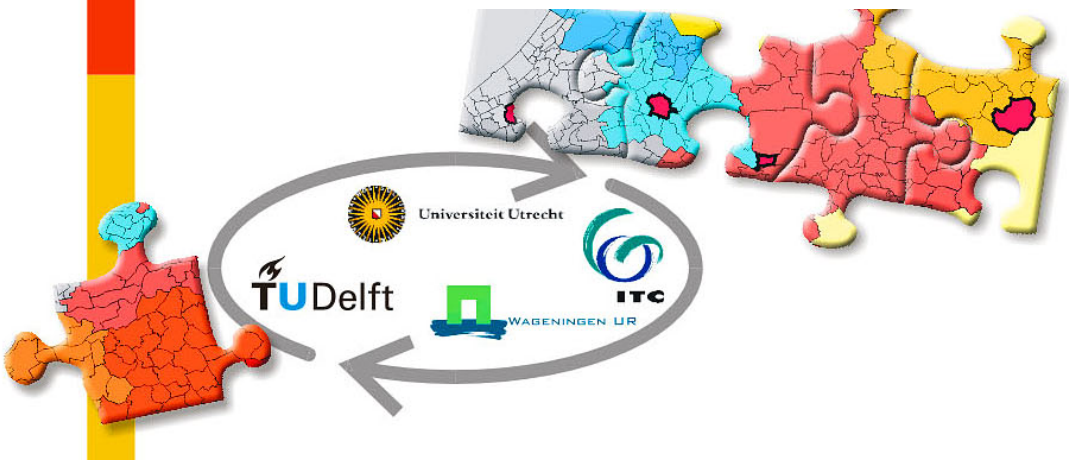
John Schaap

Wageningen, Netherlands

Professor: Prof. dr. ir. P.J.M. (Peter) van Oosterom,
OTB Research Institute,
Delft University of Technology

Supervisor: Dr. dipl. ing. S (Sisi) Zlatanova
OTB Research Institute,
Delft University of Technology

Reviewer: Dr. C P. J. M (Corné) van Elzakker
Faculty of Geo-Information Science and Earth Observation (ITC),University of Twente



Abstract

Web based multimodal travel planners are able to generate adequate travel advises for the public transport using timetable information. These travel advises are nowadays lacking detailed pedestrian route advises for the walking parts of a journey. This research focuses on developing a 3D geo data model to support pedestrian route finding for access, change and egress public transport in multimodal travel advises. The research studies the suitability of Oracle spatial 11g to find walk routes in such geo data models. It focus on geo data models to support finding an optimal route for the individual public transport traveller taking into account his specific preferences and constrains (e.g. wheelchair) for the pedestrian part of the journey in a travel advise. Travel time and time pressure, physical effort safety and conform personnel characteristics and vicinity are seen as important aspects of an optimal route. A literature study is done on existing standards and models to investigate approaches to develop such a 3D geo data model.

Based on this literature study a new conceptual geo data model is developed to support pedestrian routing in multimodal travel advises. It uses, similar as in a navigable space model (Slingsby, Raper 2008) a model representing 3D spaces only by their lower surfaces. A Stop Place Model (IFOPT 2007) is taken for it semantics describing and indentifying these spaces relevant for pedestrian routing related to public transport. To persist the geometrical aspects of these lower surfaces in a spatial database the geo data model uses simple feature (OGC 1999) surfaces. The model supports modelling of multiple layered floors and traffic areas which can be seen by a traveller on which "walking around" is possible. It also supports virtual surfaces "hidden paths" with are predefined logic connecting route areas on these floors and traffic areas. As in existing models (Lee 2004) (Stevens, Choi 2006) adjacency and connectivity of spaces are modelled with networks. The developed pedestrian model is new because attributes and behaviour of networks and surfaces are integrated. Important in this geo data model is that no difference is made between in modelling indoor and outdoor spaces. The lower surfaces representing spaces are modelled as objects inherit attributes and behaviour from nodes in a network and therefore called NodeSurfaces. Every NodeSurface is seen as possible connection for a route choice for an optimal pedestrian route. These possible pedestrian connections are modelled by combining connecting NodeSurfaces and using simple feature multiSurface to model it geometrical aspects. In this geo data model these multi surfaces are describing the potential pedestrian connections with attributes and behaviour from links in networks. Important characteristics of this multiSurface ("LinkSurfaces") are the direction (including up/down in 3D), the accessibility and pedestrian duration for a specific traveller.

A prototype implementing this new geo data model in Oracle 11g was developed. To illustrate and evaluate the new model surfaces relevant in a simple bus stop and a multimodal transport interchange where drawn in Google SketchUp and exported to KML files. These 3D surfaces where used as dataset for optimal route finding in the prototype. This research evaluates this prototype to evaluate the feasibility and suitability of the new geo data model. This prototype was also used to evaluate in suitability of Oracle spatial 11g to find walk routes in such pedestrian models. The results of the conceptual model are promising especially when modelling hidden paths. It was possible to implement the geo data model in Oracle 11g. Oracle 11g network shortest route capabilities for logical network turned out to be suitable for finding the optimal route for individual travellers with specific speed and accessibility properties. Due to the limitation that Oracle 11G does not support 3D distance calculation in geographic 3D all distances in pedestrian speed calculation were done on the geodetic surface. Optimal pedestrian routes where exported to KML files and visualisation was done with Google earth. Limitations were found for visualisation of surfaces below ground level (underpasses). The concepts of LinkSurfaces used to represent 3D pedestrian spaces in the proposed geo data models worked well in the prototype. This concept combing characteristics from links in a network and characteristic from OGC simple feature multi surfaces, is a simple, effective and well structured way to represent complex 3D spaces used by pedestrian routes in a geo data model.

Acknowledgements

First of all I would like to thank all people sharing their experiences in way finding when using public transport. Every time I told people about my research subject they starting to tell their experiences. Now I know that is possible to make a 3 hours walk to the centre of Rotterdam when taking the back entrance of Rotterdam Central station, that several people missed the tram to Nieuwegein in Utrecht because they walked to the wrong stop and that for some people the concept of front or back-entrance at the train station Ede-Wageningen is depending on where they live. It looks that public transport is a good source for way finding anecdotes.

I would like to thank the people who made it possible for me to do this research and to complete this thesis. First of all I would like to thank Sisi Zlatanova for helping me to start this research and to supervise it. It was a pleasure to have a supervisor so enthusiastic about the subject and with great expertise on 3D GIS. Peter van Oosterom for his valuable feedback and especially on modelling of node and links relations in UML. Theo Tijssen for getting all needed software and Corné van Elzakker for reviewing this thesis.

Furthermore I would like to thank to my employer REISinformatiegroep, who supported and enabled me to do the GIMA study and this research and in special Gert Staal for making good arrangements to perform this research.

Last but not least my special thanks goes to my wife Ine who have been a great source of support throughout the process of doing the GIMA study and this research and to my children Maura, Jermo and Benthe who where asked not to disturbed me when I was "always" behind the computer screen. I hope this research may help them finding their way when using public transport in the near future.

John Schaap

Bennekom, June 2010.

Contents

List of figures 5
List of tables 7
List of Abbreviations 8
1 INTRODUCTION 9
2 METHODOLOGY 11
2.1 Problem definition 11
2.2 Research objectives 11
2.3 Research questions 12
2.4 Research expectations 12
2.5 Research boundaries and limitations 13
2.6 Research strategy 14
3 EXISTING MODELS FROM LITERATURE 16
3.1 Modelling a public transport journey 16
3.2 The network space and the scene space 17
3.3 Standards for network space and the scene space in public transport 18
3.3.1 Standards for network space 18
3.3.2 Identification of Fixed Objects in Public Transport a standard for scene space 19
3.4 Standards for storage and visualisations of scene space 20
3.4.1 A standard for spatial data in a database: simple Feature Specifications for SQL 20
3.4.2 A standard for visualisation: Open GIS Keyhole Markup Language (KML) 21
3.5 3D Urban and building models 23
3.5.1 A 3D urban object model: CityGML 23
3.5.2 A building information model: Industry Foundation Classes IFC 25
3.6 3D Data models supporting pedestrian routing 26
3.6.1 Topological and graph models. 26
3.6.2 Navigable spaces model 27
4 TOWARDS A NEW MODEL FOR PEDESTRIAN ROUTING IN 3D SPACE 28
4.1 Functional requirements 28
4.2 Non functional requirements 29
4.3 Use case for pretrip journey planning to prepare for way finding 29
4.4 Scenes and scenarios to validate with a new model 31
5 A NEW CONCEPTUAL MODEL FOR PEDESTRIAN ROUTING 32
5.1 Overview of the conceptual model. 32
5.2 NodeSurface a pedestrian accessible surface 35
5.3 LinkSurface a pedestrian accessible surface with a direction 37
5.4 Modelling NodeSurfaces; "hidden paths" or "walking around areas" 39
5.5 Modelling entrances, ramps stairs, escalators and lifts with NodeSurfaces 40
5.6 Pedestrian Network and Optimal pedestrian route 42
5.7 Travellers profile 44
5.7.1 Accessibility in pedestrian route selection 45
5.7.2 Movement speed in pedestrian route selection 46
5.8 Optimal pedestrian route calculation 48
6 A PROTOTYPE FOR PEDESTRAIN ROUTING 50
6.1 Storing the pedestrian prototype data in the Oracle database. 52
6.1.1 Modelling the network and surfaces 52
6.1.2 Modelling accessibility information 57
6.1.3 Modelling walk duration 59
6.2 Operations on the pedestrian model. 61
6.2.1 Finding optimal routes 61
6.2.2 Storing optimal routes 62
6.2.3 Visualisation of optimal routes 63
7 ANALYSIS OF RESULTS 66
7.1 Evaluating the prototype on routing in scene space 66
7.1.1 Optimal route with "walk around" NodeSurfaces 67

7.1.2 Optimal route with "hidden path" NodeSurfaces 69
7.2 Evaluating the prototype pedestrian routing capabilities 71
7.2.1 Analysis of the speedtype in travellerprofiles for unrestricted travellers 71
7.2.2 Analysis of the mobility needs in traveller profiles. 72
7.2.3 Analysis of the combinations of speedtype and mobility needs in both directions 74
7.3 Experiences with database and tools used in the prototype 76
7.3.1 Evaluating Oracle Spatial usage in the Prototype 76
7.3.2 Evaluating other tools used for the prototype. 77
8 CONCLUSIONS AND OUTLOOK 78
8.1 What is essential for a 3D geo-data model to support pedestrian routing in multimodal public transport travel advices? 78
8.2 Conclusions 80
8.3 Outlook 81
REFERENCES 82
A. Applications used for prototype 84
B. IFOPT component stop place hierarchy 85
C. CityGML surface implementation in a spatial database (Kolbe, König et al. 2009) 86
D. UML diagram shown overview of IFC structure 87
E. Aspects influencing a optimal pedestrian route 88
F. Travel advice from GIMA locations Wageningen to Utrecht 89
G. Insert statement using SDO_UTIL.FROM_KMLGEOMETRY 90
H. Script to create NodeSurfaces and LinkSurfaces network in Prototype 91
I. Procedure Makeroute 94
J. SQL script to test Makeroute 96
K. Tables for network and surface 97
L. Tables for traveller profile mobility needs and speedtype 97
M. Tables for storing routing results 98
N. Table created with FME for importing surface geometries with SketchUp layer names 100
O. Views used in prototype 100
P. Google Line tool 101

List of figures

Figure 1 Research schema. 14
Figure 2 Elements of a journey using public transport with relevant actions and reasoning processes added (Rüetschi, Timpf 2005) 17
Figure 3 IFOPT sub models (IFOPT 2007)..... 18
Figure 4 IFOPT sample modelling a part of a train station (IFOPT 2007) 19
Figure 5 Point, line (String) and Polygon as basis for the simple Feature Specifications for SQL (OGC 1999) 21
Figure 6 CityGML representation showing public transport infrastructure as city furniture (Gröger, Kolbe et al. 2008)..... 23
Figure 7 CityGML LoD2 representation of a transportation complex (Gröger, Kolbe et al. 2008)..... 24
Figure 8 UML model for transportation complex (Gröger, Kolbe et al. 2008) 24
Figure 9 Boundary surfaces (Gröger, Kolbe et al. 2008) 25
Figure 10 Overview to create a 3D connectivity network (Stevens, Choi 2006)..... 26
Figure 11 Overview to create a 3D Geometric Network (Stevens, Choi 2006)..... 27
Figure 12 UML Use cases diagram for a public transport pedestrian pre-trip planning 30
Figure 13 UML Overview functional domains of the conceptual model for pedestrian routing..... 33
Figure 14 UML Overview class diagram of the conceptual model for pedestrian routing 34
Figure 15 Basic elements of the pedestrian model 35
Figure 16 Simple Feature Surface as basis for a Pedestrian Accessible surface (NodeSurface) 36
Figure 17 MultiSurface Link from NodeSurface to Ramp in green 37
Figure 18 Invalid MultiSurface Link from NodeSurface to Ramp in Red 38
Figure 19 Class diagram showing LinkSurface as multiSurface and the start and end node roles..... 38
Figure 20 Hidden path and hidden entrance as link in green 39
Figure 21 Class diagram Surfaces for visualisation and walk modelling 40
Figure 22 Lift in Google SketchUp 40
Figure 23 Lift in Pedestrian model..... 40
Figure 24 Class diagram hidden entrances, ramps stairs, escalators and lifts are NodeSurfaces with IFOPT naming..... 41
Figure 25 Example surfaces in the Google Earth to test 2D routing 42
Figure 26 Class diagram surfaces are modelled as objects inherit attributes and behaviour from nodes in a network 43
Figure 27 Example of a pedestrian network for 2D routing 44
Figure 28 Class diagram relating traveller's profile to LinkSurfaces for individual needs..... 46
Figure 29 Class diagram accessibility and speed are modelled similar..... 48
Figure 30 Flow describing the construction of the prototype..... 51
Figure 31 Detailed stop modelling by placing SketchUp models on Google earth (Bus stop Droevendaal one sides of the road direction Ede)..... 52
Figure 32 NodeSurfaces for studied location Ede-Wageningen in Google SketchUp..... 53
Figure 33 Java Classes and Interfaces for oracle Network Data Model (Murray, An et al. 2006) 54
Figure 34 Pivot SQL to show accessibility matrix..... 57
Figure 35 Class model relating accessibility and speed aspects from traveller profile to LinkSurface. . 58
Figure 36 DDL to create tables relating accessibility and speed aspects from traveller profile to LinkSurface..... 58
Figure 37 Pivot SQL to show speed matrix 59
Figure 38 FME workbench used to convert Oracle optimal routes a KML file..... 64
Figure 39 Viewing a route in Google earth as tour 65
Figure 40 Walking route between bus stop and entrance of a train. 67
Figure 41 Optimal walk route between bus stop and entrance of a train found by the prototype (using stairs down to underpass and the stairs up to platform..... 68
Figure 42 NodeSurfaces modelled as hidden paths in the underpass 69
Figure 43 Route 1: Optimal walk route between bus stop and entrance of a train found by the prototype using hidden paths for unrestricted travellers (using stairs down to underpass and the stairs up to quay)..... 71

Figure 44 Route 2: Optimal walk route between bus stop and entrance of a train found by the prototype using hidden paths for travellers restricted with medium luggage (using stairs down to underpass and the lift up to Quay 73
Figure 45 Route 3: Optimal walk route between bus stop and entrance of a train found by the prototype using hidden paths for travellers restricted by or step free access, heavy luggage or wheelchair (using ramp down to underpass and the lift up to Quay) 74
Figure 46 component stop place hierarchy (IFOPT 2007) 85
Figure 47 UMLCityGML surface implementation in a spatial database (Kolbe, König et al. 2009) 86
Figure 48 UML diagram shown overview of IFC structure (Benner, Geiger et al. 2005). 87
Figure 49 Entry webpage from www.9229ov.nl 89
Figure 50 Travel advice from www.9292ov.nl 90

List of tables

Table 1 Individual needs related to Link access types. 45
Table 2 Walk speed from literature (Lee 2005) and model values (in m/s horizontal speed) 47
Table 3 Speed profile values mapped to accessibility types (m/s horizontal speed except lift values in seconds)..... 48
Table 4 parameters for create_sdo_network 54
Table 5 Oracle SDO_NET_MEM shortest path functions (Murray, An et al. 2007)..... 62
Table 6 Example of test results output 63
Table 7 Example of pathview output (not showing the pathname and the link geometry..... 63
Table 8 Distance for every NodeSurface passed in optimal walk route between bus stop and entrance of a train 69
Table 9 Distance for every NodeSurface passed in optimal walk route between bus stop and entrance of a train using hidden paths 70
Table 10 Prototype results for walk durations for unrestricted travellers 72
Table 11 Prototype results for "normal speed" travellers in seconds 73
Table 12 Prototype route duration results bus to train scenario in seconds. 74
Table 13 Prototype route duration results train to bus scenario. 75

List of Abbreviations

2D	2 Dimensional
3D	3 Dimensional
API	Application Programming Interface
CEN	Comité Européen de Normalisation (European Committee for Standardization)
CityGML	City Geography Markup Language
COLLADA	COLLAborative Design Activity
DAE	Digital Asset Exchange
DDL	Data Definition Language
DBMS	DataBase Management System
FME	Feature Manipulation Engine
GDF	Geographic Data Files
GML	Geography Markup Language
GIS	Geographic Information System
GPS	Global Positioning System
HTML	HyperText Markup Language
IFC	Industry Foundation Classes
IFOPT	Identification of Fixed Objects in Public Transport
ISO	International Organization for Standardisation
KML	Keyhole Markup Language
LoD	Level of Detail
NDM	(Oracle Spatial) Network Data Model
MDA	Model Driven Architecture
OGC	Open Geospatial Consortium
PL/SQL	Procedural Language/Structured Query Language
PT	Public Transport
RDBMS	Relational DataBase Management System
SDI	Spatial Data Infrastructure
SQL	Structured Query Language
SDO	(Oracle) Spatial Data Option
URL	Uniform Resource Locator
UML	Unified Modelling Language
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

1 INTRODUCTION

In a journey there are places where you have to enter, change or leave transport vehicles. If you travel by car this is the place where you park your car. If you travel by public transportation these are locations such as airports, railway stations, subway stations, bus stations, ferry terminals and taxi stands. When unfamiliar with the local situation, it is usually difficult to find your way at these transport interchange places and sometimes there is stress to be on time for the next part of your trip.

Adequate travel information helps to prepare the traveller for way finding in these situations and this information is an important factor as part of the overall quality of public transport. The planning stage turns out to be the most important stage to gain travel information especially when different kinds of transport modalities are used (e.g. bus and train). In planning these multimodal public transport journeys the need for adequate information at this stage is of more importance for a public transport journey than then for a car trip (Grotenhuis, Wiegmans et al. 2007).

In journeys consisting of more than one type of transport, e.g. bus and train, a walk[1] connects the separate journey parts of the different modalities. When planning this kind of multimodal journeys, the traveller needs to take into account information provided by the various public transport timetables and the connecting walking route between the places where the traveller can enter and leave the public transport vehicles. Nowadays, timetable information is supplied over the internet. A website user selects origin and destination of a journey and the multimodal internet planner software calculates optimal public transport journeys according to the timetable information. The website user selects a proposed journey as multimodal travel advice. In existing web based multimodal internet planners a standard connecting (walking) time is taken in to consideration to connect the various transport modalities. Options to select a walking speed profile according to the individual traveller capabilities and preferences are seldom seen. Only JPL a Journey Planner for London gives the user options to plan with least walking distance, and has options for travellers with mobility restrictions e.g. avoiding stairs (Hochmair 2008). This kind of customization of multi modal travel advices is essential because the information on walk part can be as important as information on the public transport part in multi modal travel advices for individual travellers with their specific needs.

For the creation of multi modal travel advices including accurate detailed walk information in 3D on a large scale the author expects that serious amounts of geo data must be maintained[2]. For consistent maintenance of geo data spatial databases are available. The advantage of 3D topological structure management in a DBMS is that redundancy is avoided and consistency is better supported. The disadvantages are that more DBMS navigation is needed to reconstruct an object. Topology structures can be useful when many objects are neighbours of each other and share common boundaries, which have to be maintained during updates (Arens, Stoter et al. 2005). 3D models are also built with open source spatial databases and example is the GeoOxygen project using a PostgreSQL database system (Brasebin 2009). Commercial databases like Oracle 11 have spatial capabilities in combination with the capabilities for storing large networks and analyzing connectivity relations (Wang 2009). This is a graph model which is supported in Oracle 11 (Spatial). Oracle Spatial organizes graphs in four relational tables; a table for nodes, a table for links, a table for paths (routes) and table for link in these paths. Furthermore, it has the possibility to assign the real geometry to each node or link. When a space is associated as a node, the 3D polyhedron (or box) can be also stored together with this. Such a structure is proposed for calculations and visualization of 3D evacuation routes. The table for paths and the table for links in these paths are optional and can be used for storage of pre-calculated paths (routes). With the high-level languages like PL/SQL or Java the network can be built and shortest route calculations can be done (Lee, Zlatanova 2008). The suitability of these capabilities in Oracle 11 to support pedestrian routing in multimodal public transport travel advices is according to the knowledge of the author never studied or described in literature.

[1] In this document the term walk (-ing) is used for all pedestrian movement including all kinds of individual or assisted travelers locomotion by e.g. foot, wheelchair, dragging luggage, sleeping in a baby buggy.
[2] For acquisition of this data on a national scale a spatial data infrastructure (SDI) needs to be arranged.

This thesis report is structured following the approach of this research. At the start the scope and methodology of this research was defined. After this step a literature study was done to gain knowledge about existing standards and models related to geo-data models to support 3D pedestrian routing in multimodal public transport travel advices. Not one existing model was able to support directly the objectives of this research. Requirements and an evaluation approach for a new conceptual geo-data model, without the shortcomings of existing models, where defined. With these requirements a conceptual model was defined to support 3D pedestrian routing in multimodal public transport travel advices. A prototype implementation was done for clarification and to improve knowledge on the capabilities in Oracle 11 to support pedestrian routing in multimodal public transport travel advices and to evaluate the feasibility and suitability of the proposed conceptual model. After construction of this prototype it was validated and the results where analysed. At the end of this research the thesis report was written including the findings of this research. This thesis report documents the research, after its first introduction chapter, in the same order:

Chapter 2 defines the scope and methodology of this research in detail. It focuses on theme, objectives, research questions and approach to perform this research on geo data model to support 3D pedestrian routing in multimodal public transport travel advices.

Chapter 3 describes the findings of the literature study. It focuses on existing standards and models found in literature related to public transport, 3D spatial data models and pedestrian routing.

Chapter 4 contains the requirements and cases to evaluate a geo data model to support pedestrian routing in 3D space to be used in multimodal public transport travel advices.

Chapter 5 is the description of a new proposed conceptual model to support pedestrian routing in 3D space. This chapter describes the elements of this new conceptual model in detail. The Unified Modelling Language (UML) is used as formal description of the proposed geo data model in this conceptual model.

Chapter 6 focuses on the prototype implementation of the geo data model. The storage and usage of the elements of the proposed model in the prototype using an Oracle spatial database is described in detail.

Chapter 7 analyses the prototype. It analyses the capabilities of prototype implementation and the capabilities of Oracle spatial to support pedestrian routing in 3D space in multimodal public transport travel advices.

Chapter 8 describes the conclusions of this research and gives an outlook to support pedestrian routing in 3D space in multimodal public transport travel advices.

2 METHODOLOGY

As mentioned in the introduction at the beginning of this research the scope and a strategy to follow in this research was defined. This chapter first describes this scope, a description is given of the underling problem this research studies and defines the research objectives and questions. The scope is further defined by describing expectations and limitations of this research. The strategy use in this research is described and visualised in a research schema (figure 1). This strategy is using prototyping to do empirical analysis on the research questions. The strategy used in this research can be classified as a combination of essential prototyping and (design) technical prototyping (Reeve, Petch 1999).

2.1 Problem definition

When a potential traveller plans a multimodal journey he is using information of public transport network and its timetable for the part of the journey where he takes public transport. Multi modal travel planners are able to generate adequate travel advices[3] for the public transport part of the journeys but for are not taking the geometry of walking part of a journey into account. Therefore these travel advices are nowadays lacking detailed 3D walking route advices for the walking parts of the journey. To improve this situation it is critical to have an effective and well-structured geo-data model for that can be used by 3D routing methods.

This research considers the hypothesis that the influence of the geometry of the access, interchange or egress walking part of a travel advice can not be neglected in a multimodal public transport advice for an individual traveller. It questions that in most journey planners nowadays standard connection (walking) durations, based on a fixed walk speed and distance, are used for all different connections at one public transport interchange place. The needs of the individual travellers (with possible access or mobility restrictions) and the specific geometry of walking areas for a specific connection makes that no 'standard' connection time exists for such an advice. The fact that no explicit directions for travellers (walking routes) are given at the moment in these travel advices taking the third dimension of the geometry of walking areas into account is seen as a major shortcoming. Realistic 3D visualisation showing walking directions added to travel advices gives the traveller the opportunity to prepare himself for a walk in an unknown situation. To explore the modelling and implementation difficulties and to create these explicit directions in 3D space an important goal of this research is also to explore the capabilities of a DMBS (Oracle 11) on storing this geometry and performing 3D routing in the geometry of the walking parts in individual travel advises.

2.2 Research objectives

To research the stated hypothesis with the intention to improve journey planners in the future this research focuses on the design of a effective and well-structured geo-data model for that can be used by 3D routing methods and is implementable in a DBMS, the following 2 research objectives are defined for this research:

- *Design, develop and evaluate a geo-data model to support access, interchange and egress walk-routing in multimodal travel advices.*
- *Research the suitability of Oracle 11 to find walk routes in 3D geo-data relevant for the walk part of multimodal public transport journeys.*

[3] In this document 'travel advice' is used instead of 'itinerary' to emphasize that a travel advice contains integrated multimodal travel information and the planned route (itinerary) is just a part of this information.

2.3 Research questions

Based on the described research objectives the following research questions and sub questions are defined:

1. Which requirements can be defined to a geo-data model supporting access, interchange and egress walk-routing in multimodal travel advices?

Sub questions:

- Which use cases (traveller's scenarios) the geo-data model must support?
- Which requirements the geo-data model must support?
- Which standards to consider in this geo-data model?

2. How to model the 3D space to support access, interchange and egress walk-routing in multimodal travel advices?

Sub questions:

- Which geographical elements are relevant to a traveller in the access, interchange and egress walking route?
- What is needed to model the 3D and topological relations of these elements into a geo-data model?
- What kind of network can model routing for travellers between these elements?
- What distinguish routing for travellers having access or mobility restrictions?

3. Is a database supporting topological network models like Oracle 11 suitable for supporting a geo-data model for access, interchange and egress walk routing?

Sub questions:

- Are data storage models in Oracle 11 adequate for this purpose?
- Does Oracle 11 support constructing a 3D geometric network from a 3D building model?
- Are shortest route capabilities supplied by Oracle 11 suitable for finding walking routes to support access, interchange and egress walk-routing in multimodal travel advices?

2.4 Research expectations

Innovation in the field of modelling walk routing to connect to public transport is expected as outcome of this research. This research can influence the geo models used in future travel planners to support 3D public transport and pedestrian routes in planning algorithms. It is also possible that additional (web) services will arise enhancing the travel advices of existing web planning services. The research outcome is not limited to walk routing around transport places. New applications and research in other areas for way finding and 3D navigation could benefit from this research. General knowledge about data modelling for walk routing can help science a step forward in this area. The results from this research could be useful when developing new travel planners and navigation systems in the future. Furthermore extended knowledge in the research area can influence conceptual models used in spatial data infrastructures.

2.5 Research boundaries and limitations

This research is limited to the research of geo-data models in relation to described objectives. It focuses on 3D geo-data models to support pedestrian routing in multimodal public transport travel advices therefore this research studies:

- Requirements and standards for 3D geo data models to support pedestrian routing
- Modelling pedestrian accessibility in 3D geo data models
- Route planning for walk routing over surfaces in a 3D space
- Storage and usage of 3D geo data for pedestrian routing
- Topological network and 3D capabilities of a spatial database (Oracle 11G)

Outcome of this research might be useful for other research but explicit out of scope of this research are:

- Modelling public transport
- Modelling road/street networks
- Modelling scene space street environments
- Route planning on the public transport network
- Route planning on the road/street network
- Movement planning in cross country
- Searching timetables
- Mathematic shortest route algorithm research
- Usability study
- Symbolisation geographical elements relevant to a traveller
- Optimizing visualisation
- Modelling travellers flows
- Location allocation modelling
- Route tracing
- Location based services
- Personal navigation devices
- Non-spatial database capabilities
- Comparison of database systems
- Database performance tuning
- Data collection and exchange
- Spatial data infrastructures

The timeframe of this research was limited to the academic year 2009-2010. A selection is made for public transport places to be studied. Three public transport places where candidate to implemented in the prototype of this research. At first a simple public transport place consisting of two bus stops at each side of the road to model access and egress for the modality bus in 2D. Second a model containing one bus stop and a train station to focus modelling on multimodality and 3D accessibility (including underground pedestrian passageway, ramps, stairs, elevator, pavement and platforms). A third complex public transport place of national importance containing a complex train station, a complex city bus station and a regional bus station in combination with a local tram stop was not implemented in the prototype because of the limitation of the timeframe.

2.6 Research strategy

As approach for this research the elementary GIS steps, exploration, analyses, synthesis and presentation are used as fundamental concept for meeting the research objectives. This approach has led to the research schema as shown in Figure 1. In this schema the activities of this research are mentioned. This schema can be used as overview when reading this thesis; references are made to relate the contents of this thesis to the activities in the research.

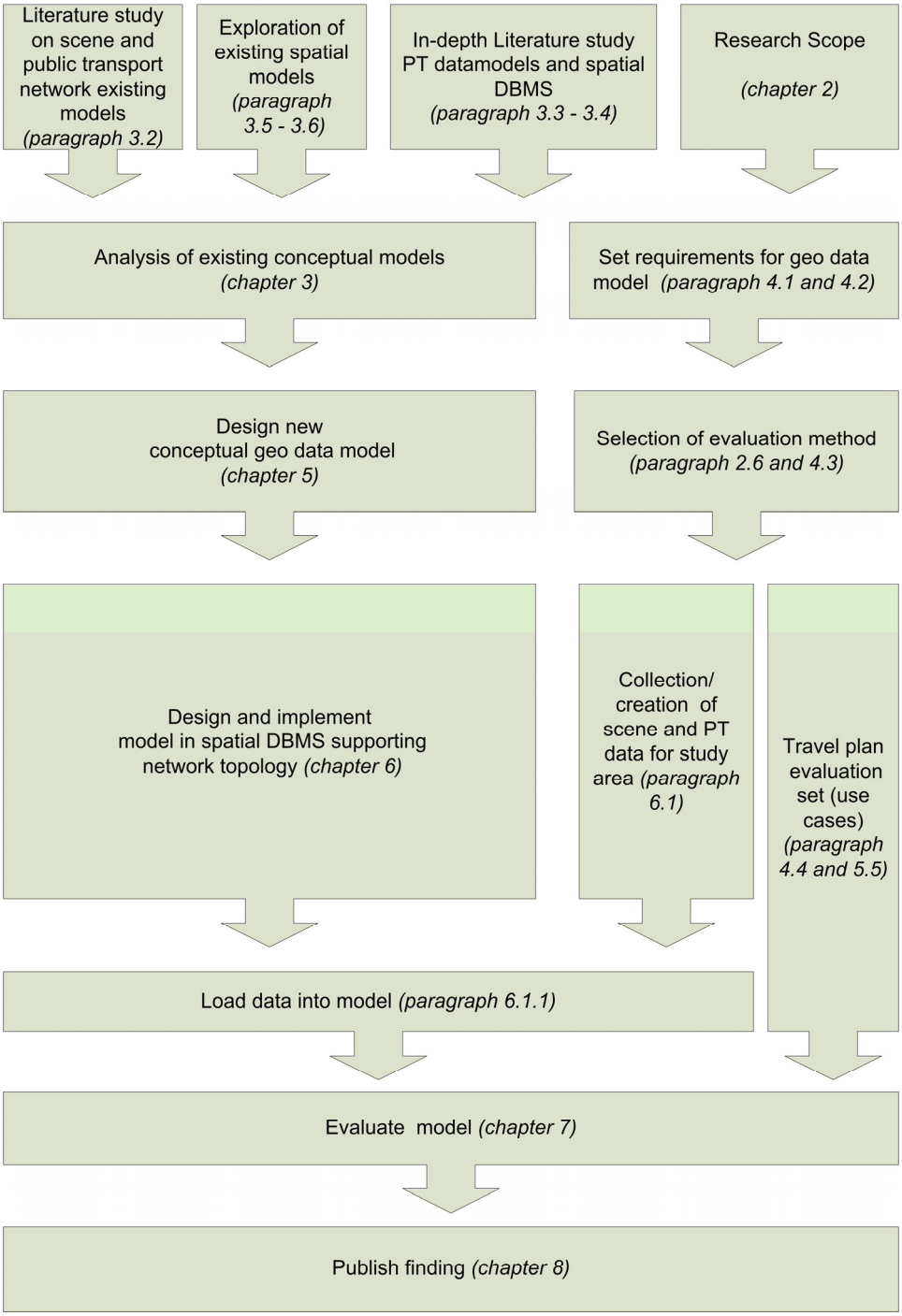


Figure 1 Research schema.

To explore the problem area a literature study was started. This exploration was done to find existing conceptual models from the field of public transport modelling focusing on the combining of the walk part (scene) and the public transport (network) in a travellers journey. Other literature was studied, knowing that the pedestrian part of a journey does can cover indoor and outdoor 3D space. This studied literature did focus on exiting 3D spatial models and spatial models related to pedestrian routing. In depth exiting public transport standards where studied and in preparation to develop a prototype literature focusing on the capabilities of a spatial database (Oracle) where studied. In this research knowledge was gained from various sources. The library of the University of Utrecht was used as main source to peer reviewed scientific literature. Other online sources (internet) were used for finding information standards as described in the references. Access to these sources was arranged and verified before the start of this research. Information needed for construction of the prototype in Oracle 11G will was gained from the Oracle manuals and free online sources.

After the initial explorative literature study a project proposal was written and agreed to scope this research by defining theme, objectives, research questions and approach. Furthermore procedural agreements where made on planning, supervision and contact hours. A set of requirements for an effective and well structured geo-data model to support 3D pedestrian routing in travel advises where specified by the author based on the agreed research scope. The existing models found in literature were documented and analysed. Existing models were evaluated for it suitability to support 3D pedestrian routing in public travel advises.

Based on the findings of the previous steps a new conceptual model to support pedestrian routing in 3D space was developed. The Unified Modelling Language (UML) was used as formal description of this proposed geo data model in this conceptual model.

A prototype approach was selected for evaluating the proposed conceptual model. The set of requirements were used as basis to select locations and traveller scenario's (use cases) for evaluation of this new conceptual model. The proposed conceptual model was the basis for a prototype implementation of a geo-data model in Oracle 11. This approach can be classified as a combination of essential prototyping and (design) technical prototyping (Reeve, Petch 1999) This approach was used for clarification and to improve knowledge on the objectives of this research and to evaluate the feasibility and suitability of the proposed conceptual model. Datasets for selected public transport locations were composed. Sample multimodal travel advices were extracted from the www.9292ov.nl website. Scene data about the selected public transport locations was extracted manually from free online resources (www.ns.nl and maps.google.com). These dataset were loaded in the prototype. The main purpose of the data was to illustrate and validate the proposed geo-data model not to have up to date, accurate data about existing infrastructure.

To evaluate the conceptual geo data model, the prototype was analysed for the selected traveller scenarios. The prototype is a used for finding optimal routes for these scenarios. These results were analysed and documented. At the end of this research this thesis report was composed out of the results of the previous steps and the findings of this research were written.

3 EXISTING MODELS FROM LITERATURE

In literature models are found from different domains which are related to geo-data models to support pedestrian routing in multimodal public transport travel advices. Some of these models are having their background in modelling public transport and are focussing on the public transport network and its environment. Other models are focusing on the modelling of 3D space and are modelling indoor or outdoor environment for other purposes. An effective and well structured model of the 3D space is needed to support walking routing. In the field of routing for emergency services (Lee, Zlatanova 2008) a categorization is made of 3D geometric models, 3D topological and graph models and 3D city models and 3D Building models. All models are designed for there own purpose. Some models are focusing on storing geometrical data while others on visualisation of this data, some models are a strong on the semantic aspects, others on supporting navigation. All these models are having there own standardisation and language to store and interchange data. No model was found to support pedestrian routing in multimodal public transport travel advices in 3D directly, therefore relevant elements and representations of these models where studied as source for a new 3D geo-data model to support pedestrian routing in multimodal public transport travel advices. In this thesis the studied models are categorized as models focussing on to public transport, models for storing and visualisation of 3D data, 3D urban and building models and models supporting pedestrian navigation. This categorization is used to describe the models. In paragraph 3.1, 3.2 and 3.3 models and standards related to public transport are described. They are mentioned to develop a conceptual idea how to combine the traveller, the walk part of journey and the public transport part of a journey in travel advises. In paragraph 3.4 standards are describe to store and visualize 3D data, paragraph 3.5 is focusing on the indoor and outdoor urban and building models and paragraph 3.6 on 3D data models supporting pedestrian routing.

3.1 Modelling a public transport journey

In this research the walk part of a transport journey is seen as integrated part of a public transport journey. Therefore the model for a walk part must be integrated in a model for public transport. The walk part of a public transport journey exists in 3 parts of a journey, when accessing the system, transferring and leaving the system (Rüetschi, Timpf 2005). Figure 2 shows the interaction between these 3 parts.

Furthermore public transport is often organised according to the line operation principle, it combines four entities:

1) A network of roads, contact wires, or tracks;
2) Public transport lines, which use these roads, track, or contact wire network;
3) Stops
4) A timetable.

These entities are restricting the walk part in a public transport journey. This model is used to define requirements for pedestrian routing in this research.

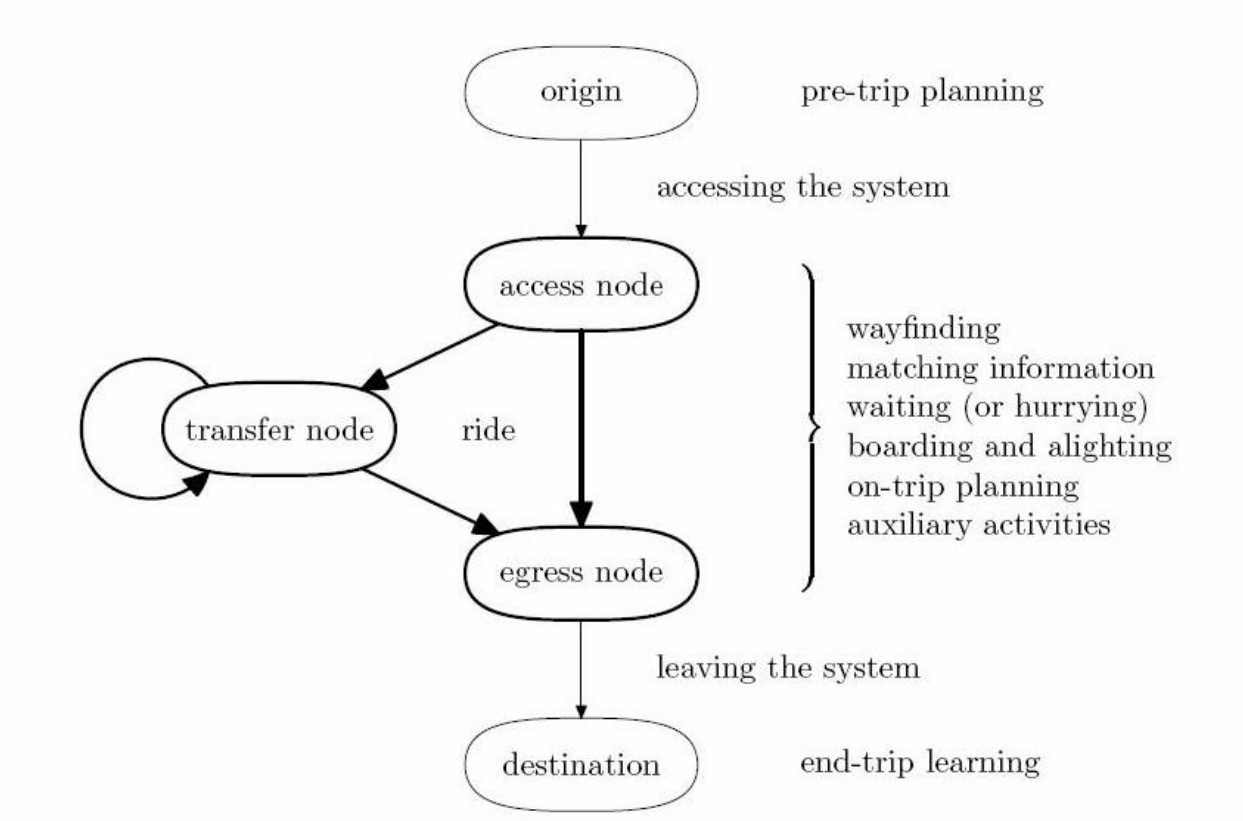


Figure 2 Elements of a journey using public transport with relevant actions and reasoning processes added (Rüetschi, Timpf 2005)

3.2 The network space and the scene space

A difference exists between the environments a traveller experiences when he is using a vehicle of a public transport provider and when he is walking during an interchange (Rüetschi, Timpf 2005, Rüetschi 2007). In the domain of public transport he experiences a (public transport) network. This environment is dominated by connections made by transport lines (links) and the locations (nodes) where you can enter and leave or change the vehicles. It's the world of timetables, line maps, fare structures, alternative journey routes and sometimes delays. The public transport environment a traveller experiences is called network space by Rüetschi because it is based on the public transport node/link structure. This is the world created by strategic planners of public transport and the result of network design, line planning, and timetabling (Borndörfer, Grötschel et al. 2006). This world is designed to optimize travel times for the public and it is influenced by historical, social, and economical processes. The environment a traveller experiences when he walks has a different origin, it created by architectural and urban design. This walk environment is called a scene space; it is an environment for way finding that has another structure and is dominated by nested open spaces. Environments like stations (with its halls, squares, platform areas, etc.), shopping malls, or public parks are samples of these walk environments. How these spaces are nested and connected to each including possible level changes by stair or elevators is needed for orientation and way finding. Detailed information to prepare travellers for navigating in these environments is hardly available over the Internet (Rüetschi, Timpf 2005).

The scene space a traveller experience when walking is a 3D space. In this space the traveller is only expected to walk in allowed areas according to allowed routes. All his movements in this space are restricted by spatial and temporal constraints established by the public transport network space and its public transport schedule. Under different circumstances (e.g. emergency situations) and with specific privileges (e.g. allowance to first class traveller waiting rooms, ticketing restrictions) these access rules

can change. Furthermore capabilities of the traveller can restrict these routes (e.g. wheelchair accessibility).

3.3 Standards for network space and the scene space in public transport

3.3.1 Standards for network space

For the network space a Europeans standard for in the exchange of information is TransModel (CEN WG3 TC278 2001). In this model the public transport network is defined in a standard way and exchange of passenger information is strongly influenced by this because basically all types of passenger information generally use many elements of this topological network to describe the timetables and stop locations and related information. TransModel has a reference data model describing public transport services with point and links. This model is also the origin of Dutch public transport interchange standards called, TransModel Interface (TMI) standards. Another industry standard for network space and used by web based public transport planners is "The Google Transit Feed Specification" (Google 2009b). This specification is less detailed than TransModel. Model wise the core of the network model is similar because all the elements of this specification can be mapped to TransModel entities (Knowles 2008).

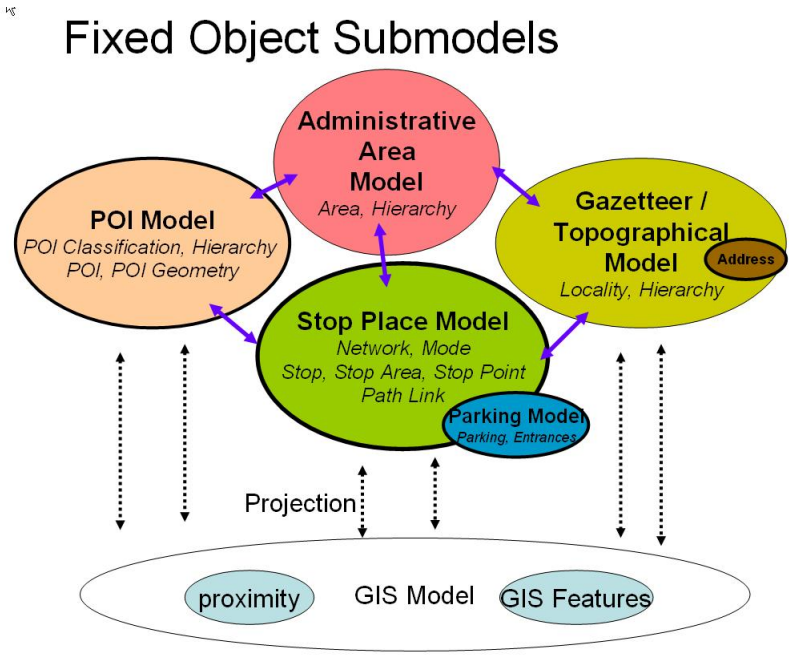


Figure 3 IFOPT sub models (IFOPT 2007).

3.3.2 Identification of Fixed Objects in Public Transport a standard for scene space

For scene space in public transport a standard for describing these spaces is a CEN standard in development (IFOPT 2007). This standard "Identification of Fixed Objects in Public Transport" models transfer locations. It consists of 5 sub models as shown in Figure 3. The pedestrian elements of used in this research the scope are defined in the Stop Place Model The Stop Place Model is a data model to describe the structure of an interchange including entrances, pathways, and accessibility limitations. The Stop Place Model indentifies spaces, called fixed objects in this standard, relevant for pedestrian navigation. Figure 4 illustrates the fixed objects in a train station. A red dotted line specifies a possible path between a specific boarding position and fixed location in a hallway (IFOPT:Accesspace). It defines the problem of pedestrian routing in interchange locations as follows *"5.2.13 JP#13 To provide detailed guidance for making an interchange between two services over a connection."*

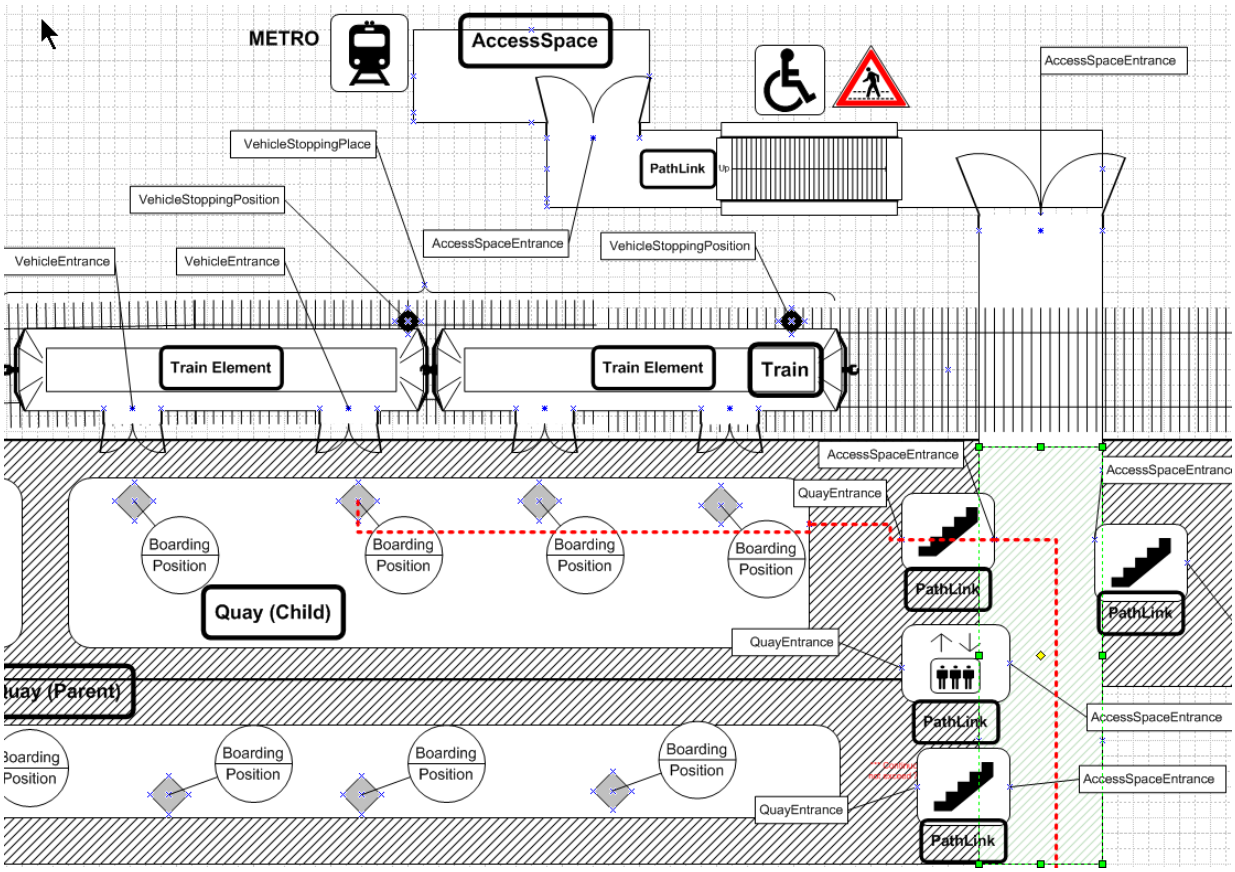


Figure 4 IFOPT sample modelling a part of a train station (IFOPT 2007)

The Fixed Object Model has a relationship to other standards describing the geographical features of a country, but is not itself a GIS standard. The Fixed Object Model describes the logical structure of stop places in a way that can be related to the Public Transport universe of discourse of TransModel. TransModel and the Fixed Object Model exclude the detailed description of geographic features, both models are having capabilities to project objects to a GIS model. The Location model used in the Fixed Object Model is represented in a way such that it can be projected onto a variety of spatial representations. Coordinate reference systems are not defined in these models and can be defined when implementing this projection to a spatial representation. The Fixed Object Model upholds the principles from TransModel of separation of information layers and the use of Point and Link representations within the distinct layers which can be used to project elements between the models. For 2D representation Point and Link and address data items are included. Transport interchanges are often complex buildings with many interconnected levels, the Stop Place Model support vertical levelling naming. Level is a distinct concept from that of a vertical spatial coordinate and is a semantic

label. Example of these semantic labels could be "Basement", "Floor 1" or "Underpass". Geographic capabilities to define surfaces connecting these vertical levels in 3D are lacking. The IFOPT Stop Place Model is strong in defining the semantic aspect, organizing, indentifying and naming, of spaces relevant to support pedestrian navigation in relation to public transport areas.

In the Stop Place model the following entities are considered relevant to travellers**:**

- **Passenger**:
  - Quays, Boarding Points,
  - Entrances
  - Access Spaces
  - Paths between Nodes
- **Vehicle**:
  - Tracks, Stopping positions
- **Equipment**
  - Ticket machine, lockers,

In the Stop Place Model semantic elements exists to describe:

- Spaces which are walk able and publicly accessible
- Spaces (stores, kiosks, public toilets)
- Buildings/ part of buildings and indoor spaces
- Connecting elements between levels (steps, escalators, elevators, stairways ramps)

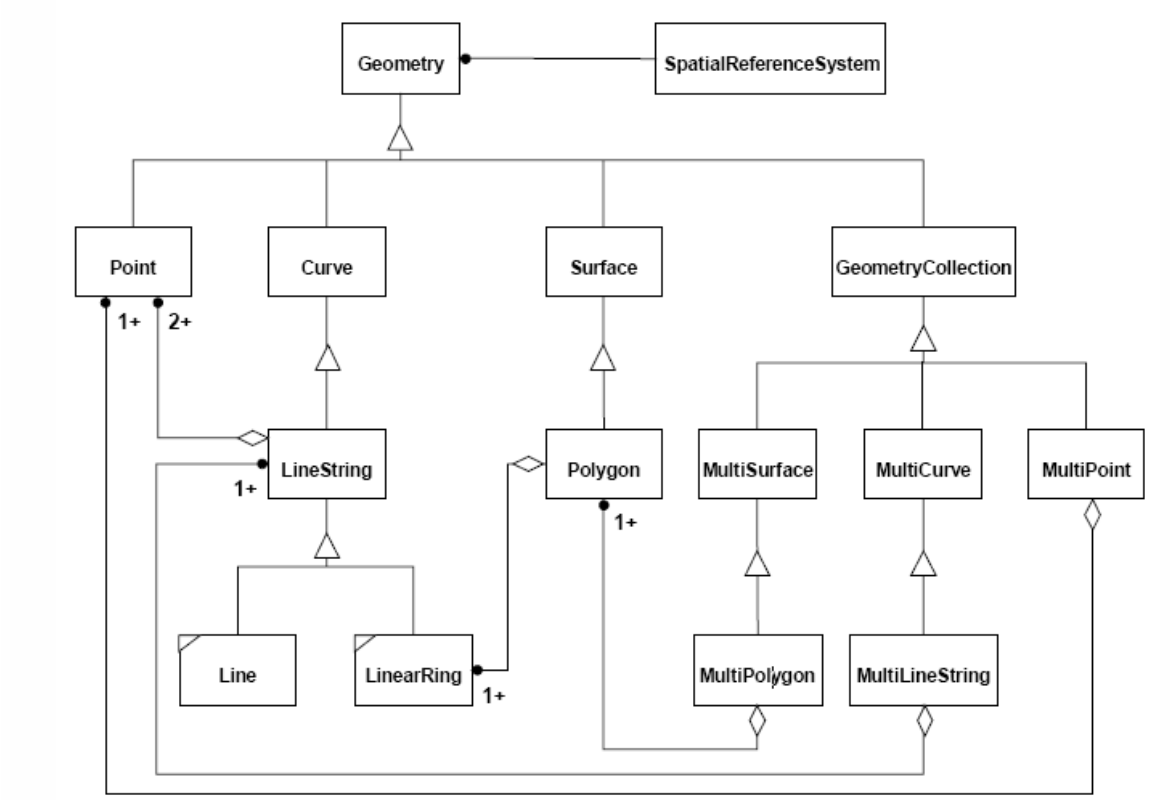
A weak point for the Stop Place Model is that it is a complex standard with a lot of classes. A "light" version of the standard which could be extended with local relevant elements would be easier to implement. However inn the complex standard not all relevant items for travellers are included. Relevant items for travellers are signs. These signs are used for orientation. The location of these signs in relation to the footpath of the traveller must be in line with the descriptions of directions used in travel advises. Capturing all information from signs and there locations and direction in relation to the footpath will improve the information in travel advises, but signs are not included in the IFOPT standard.

3.4 Standards for storage and visualisations of scene space

3.4.1 A standard for spatial data in a database: simple Feature Specifications for SQL

The basis elements of the geometric models in 3D data models are build on the datatypes points, lines and polygons. These elements are seen in the Simple Feature Implementation Specification for SQL (OGC 1999), which provides guidance for implementing spatial data types in Database Management System (DBMS) Figure 5 shows the UML diagram for this specification. The scene a pedestrian experience consists of 3D objects. These real world 3D objects can be modelled with object (called features), representing real world objects having spatial and nonspatial attributes. 3D objects can be represented as polyhedrons in different ways. From the Simple Feature Specification for SQL the data type polygons or data type multipolygon can be used. Using the polygon approach, one or two more columns have to be introduced a relational table, o specify that a polygon belongs to a particular 3D objects. One 3D object is represented by several rows in the geometry table. In the multipolygon approach, a 3D object is described in one row, since all the information about the polygon is decoded in a spatial geometry type. In this case the number of records is reduced, but redundancy of coordinates cannot be avoided. Each triple coordinates is repeated at least three times in the list of coordinates. An apparent advantage of the 3D multipolygon approach is the one-to-one correspondence between a record and a 3D feature (Lee, Zlatanova 2008). A limitation of the Simple Feature Implementation Specification for SQL is that it does not have a data type to represent polyhedrons as single objects. Oracle 11g has a data type solid (SDO_GEOMETRY with SDO_TYPE = 3008) and multisolid (SDO_GEOMETRY with SDO_TYPE = 3009) to represent polyhedrons (Murray 2010). A solid can be used as direct representation of a polyhedron. A solid is defined as an object

having multiple surfaces and these multiple surfaces are completely enclosing this object in a three-dimensional space. An object of type multisolid can represent a collection of solids in one single object. The redundancy of coordinates and reduction of records does also exist as described for the multipolygon approach. The advantage is that one complex 3D feature consisting of one or more solid object can be stored in one record. In this way one multisolid object can represent one complex 3D feature modelled with one or more polyhedrons in a single record



.

Figure 5 Point, line (String) and Polygon as basis for the simple Feature Specifications for SQL (OGC 1999)

3.4.2 A standard for visualisation: Open GIS Keyhole Markup Language (KML)

Keyhole Markup Language (KML) is XML language for representing geographic information. KML was originally developed by Keyhole, Inc. and used in the Google Earth viewer, which at the time was named Keyhole Earth viewer. In 2008, KML is officially approved by the Open Geospatial Consortium as an Open Standard (OGC 2008). KML is used to encode and transport representations of geographic data for display in an earth browser. KML encodes what to show in an earth browser, and how to show it. Geographic visualization includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look. KML is an XML language focused on geographic visualization, including annotation of maps and images. Geographic visualization includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look. From this perspective, KML is complementary to most of the key existing OGC standards including GML (Geography Markup Language), WFS (Web Feature Service) and WMS (Web Map Service). Currently, KML 2.2 utilizes certain geometry elements derived from GML 2.1.2. These elements include point, line string, linear ring, and polygon similar elements as found in the Simple Feature Implementation Specification for SQL (OGC 1999). For representing polyhedrons extruding elements is possible. A polygon representing a surface can be extruded to represent a polyhedron. This functionality makes is easy to represent for example a simple shape of a building by the roof only. The extrude functionality can be used to connect the geometry of the roof to the ground to make the simple 3D representation

of a building. Another approach of representing polyhedrons in KML is using a MultiGeometry datatype. A MultiGeometry is a collection containing zero or more geometry elements. To represent a polyhedron a MultiGeometry can be used containing all relevant polygons for this polyhedron. A KML file may include a variety of different features, for example polygons, images, 3D models and place marks. For positioning, KML uses 3-dimensional geographic coordinates containing longitude, latitude and altitude.

KML has a lot of features which can be useful to visualize objects relevant for travellers planning there public transport trip:

- Annotate the earth
- Specify icons and labels to identify locations
- Create different camera positions to define unique views
- Define styles to specify KML feature appearance
- Define image overlays to attach to the ground
- Write HTML descriptions of KML features
- Define location and orientation of textured 3D objects

Using this features the semantics (identifying and describing the purpose with attributes) of the objects relevant for a traveller can be expressed combining visual and textual information. The possibility to organize KML features into hierarchies give the possibility to describe the semantically important nesting of open spaces as described in paragraph 3.2 to a traveller. KML is mainly used for visualisation on maps representing objects or routes in the outdoor environment and can be used to create overlays for existing Google maps. The overlays can include detailed 3D model information with textures, and polygons which represent 3D geometry of objects. In OGC KML user extension can be defined. The KML format is used by Google Earth and has the possibility to visualize semantic and geometrical aspects of objects relevant for pedestrian routing for this research. To visualize a route in an Earth browser KML has the feature (extension) of touring (Google 2009a). To include 3D models in KML files an extension is made by Google to add COLLADA formatted data (Google 2009c). KML files are offended distributed as (ZIP) compressed files with a file extension "KMZ". The 3D drawing tool Google SketchUp can be used to create these 3D models in this "KMZ" file format. Files of this format (KMZ files) can be used as source for an import into a spatial database.

For pedestrian routing altitude is relevant for travellers. Altitude is defined in KML with a feature called kml:altitudeMode. This altitude mode can have in OGC KML two values:

1) relativeToGround - Sets the altitude of the element relative to the actual ground elevation of a particular location. For example, if the ground elevation of a location is exactly at sea level and the altitude for a point is set to 9 meters, then the elevation for the icon of a point placemark elevation is 9 meters with this mode. However, if the same coordinate is set over a location where the ground elevation is 10 meters above sea level, then the elevation of the coordinate is 19 meters. A typical use of this mode is for placing telephone poles or a ski lift.
2) absolute - Sets the altitude of the coordinate relative to sea level, regardless of the actual elevation of the terrain beneath the element. For example, if you set the altitude of a coordinate to 10 meters with an absolute altitude mode, the icon of a point placemark will appear to be at ground level if the terrain beneath is also 10 meters above sea level. If the terrain is 3 meters above sea level, the placemark will appear elevated above the terrain by 7 meters. A typical use of this mode is for aircraft placement.

From a travellers point an altitude "relative to start location of pedestrian walk" is of relevant interest. The vertical (attitude) change during a walk is of importance when planning a walk especially for travellers with mobility restrictions.

3.5 3D Urban and building models

3.5.1 A 3D urban object model: CityGML

CityGML is a common semantic information model for the representation of 3D urban objects that can be shared over different applications (Gröger, Kolbe et al. 2008). CityGML is an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is implemented as an application schema of the Geography Markup Language 3 (GML3), the extendible international standard for spatial data exchange and encoding issued by the Open Geospatial Consortium (OGC) and the ISO TC211. CityGML is based on a number of standards from the ISO 191xx family, the Open Geospatial Consortium, the W3C Consortium, the Web 3D Consortium, and OASIS. CityGML defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantic, and appearance properties. "City" is broadly defined to comprise not just built structures, but also elevation, vegetation, water bodies, "city furniture", and more. Included are generalisation hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. CityGML is applicable for large areas and small regions and can represent the terrain and 3D objects in different levels of detail simultaneously. Since both simple, single scale models without topology and few semantics or very complex multi-scale models with full topology and fine-grained semantically differentiations can be represented; CityGML enables lossless information exchange between different GI systems and users.

Figure 6 CityGML representation showing public transport infrastructure as city furniture (Gröger, Kolbe et al. 2008)

In CityGML Outdoor public transport infrastructure is modelled as "City furniture" and traffic areas. "City furniture" objects are immovable objects like lanterns, traffic lights, traffic signs, flower buckets, advertising columns, benches, delimitation stakes, or bus stops.

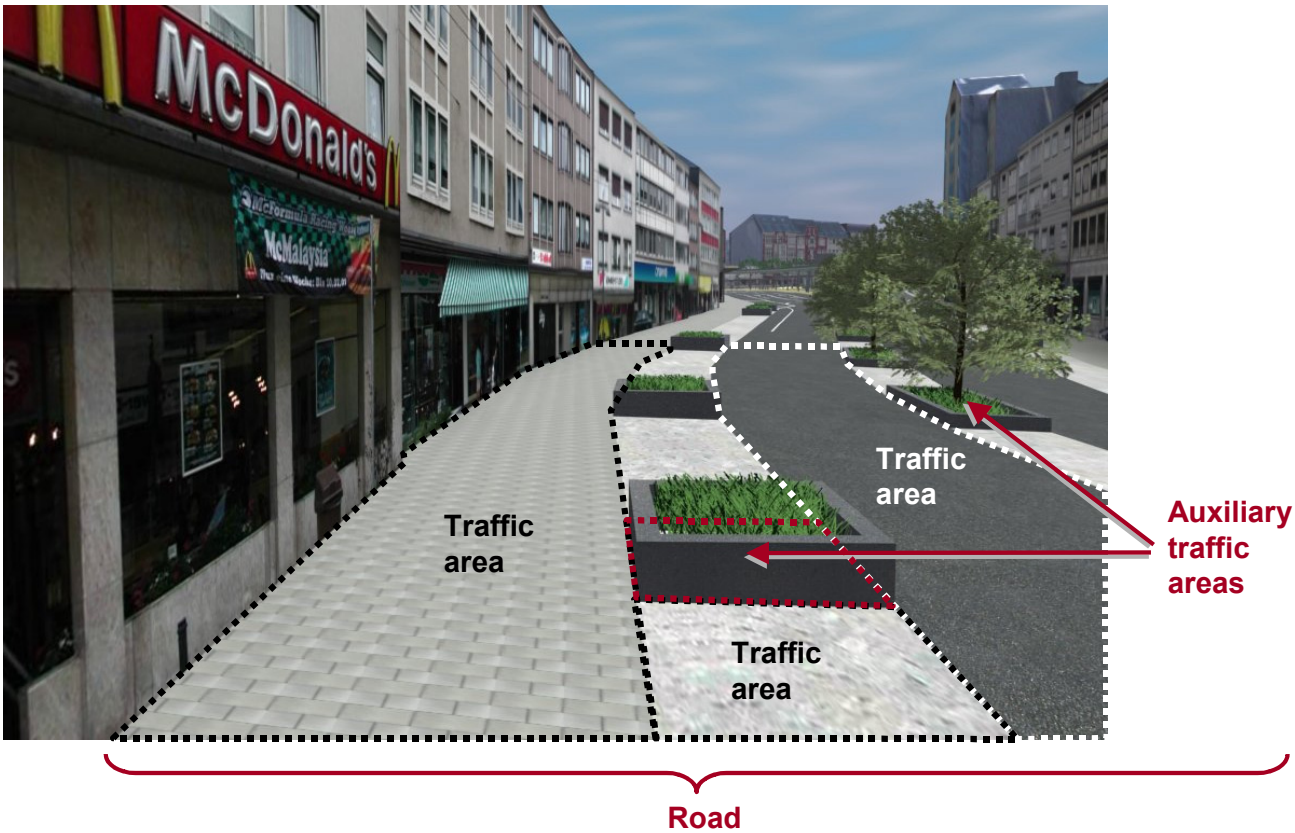


Figure 7 CityGML LoD2 representation of a transportation complex (Gröger, Kolbe et al. 2008)

Transportation features are represented as a linear network in the lowest level of detail (LoD0). In more detailed representations, starting from LoD1, all transportation features are geometrically described by 3D surfaces. The TrafficAreas are those elements, which are important in terms of traffic usage, like car driving lanes, pedestrian zones and cycle lanes. The AuxiliaryTrafficAreas are describing further elements of the road, like kerbstones, middle lanes, and green areas.

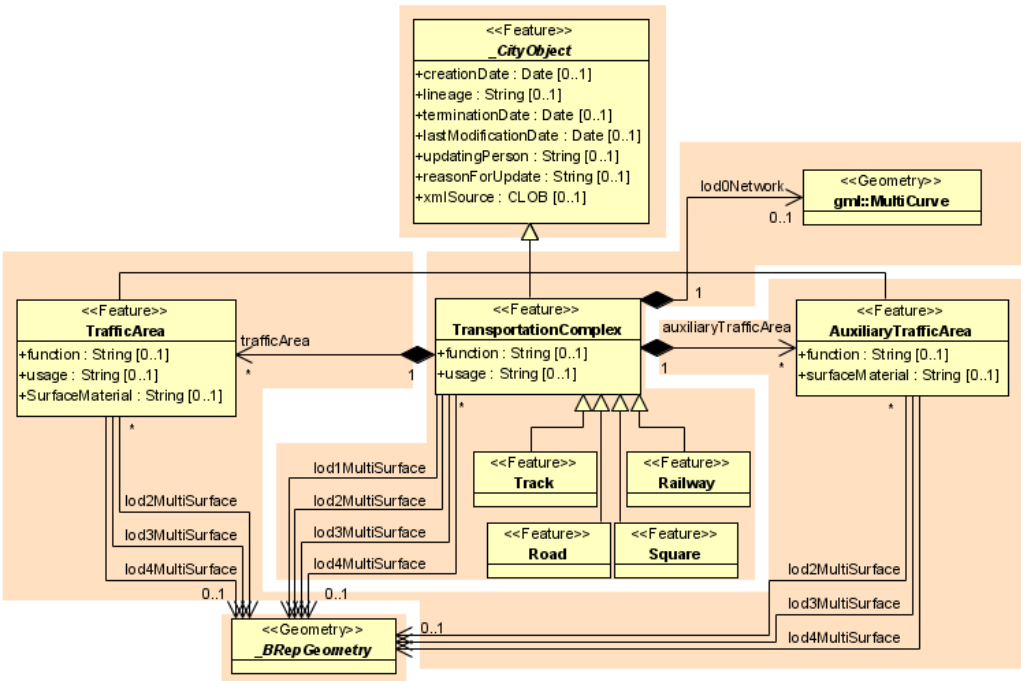


Figure 8 UML model for transportation complex (Gröger, Kolbe et al. 2008)

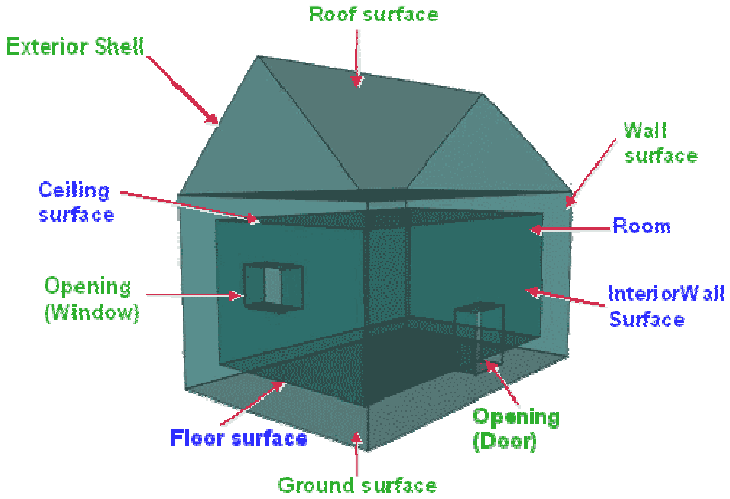


Figure 9 Boundary surfaces (Gröger, Kolbe et al. 2008)

Most existing CityGML models are mainly focussing on modelling outdoor 3D urban areas, but the standard can also support indoor 3D. Indoor pedestrian areas are called floor surface in CityGML. A visible surface of a room is represented geometrically as a Solid or MultiSurface. Semantically, the surface can be structured into specialised BoundarySurfaces, representing floor (FloorSurface), ceiling (CeilingSurface), and interior walls (InteriorWallSurface) (Gröger, Kolbe et al. 2008). The standard is lacking detailed semantic objects relevant for pedestrian navigation for public transport. CityGML is extensible and adding these elements could form a good basis for pedestrian navigation related to public transport.

3.5.2 A building information model: Industry Foundation Classes IFC

A Building Information Model (BIM) is a digital representation of physical and functional characteristics of a facility. Industry Foundation Classes are a standard in this field. IFC represents a data schema for sharing construction and facility management data across various applications used in the building construction industry (buildingSMART). It is a common exchange format in the field of architecture, engineering, and facilities management. It is an object-oriented data schema based on class definitions representing the objects (such as building elements, spaces, properties, shapes, etc.) that are used by different software applications used in construction or facility management project (To illustrate this object-oriented schema in appendix D a UML diagram with an overview of the main IFC classes is given). IFC is a standard for building information models, not for drawings. It enables to exchange information about building structures, elements, spaces and other objects between systems for 3D and 2D shape, properties and attributes, parameters and relationships (e.g. connectivity) among elements are examples for the content of IFC. IFC is an open specification, supported by an international, non-for-profit organization and it is registered with ISO as ISO16739. It is not owned by a single vendor or group of vendors. Research on mapping elements from IFC to CityGML exists (Benner, Geiger et al. 2005). Building models are focussing on the indoor environment and on the construction details of facilities they can contain strong detail on the geometrical details of these elements. Although only a small subset of these details might be relevant for pedestrians, they are mentioned in this research because they could be a good basis for automatic subtraction of these details to fill future pedestrian models.

3.6 3D Data models supporting pedestrian routing

3.6.1 Topological and graph models.

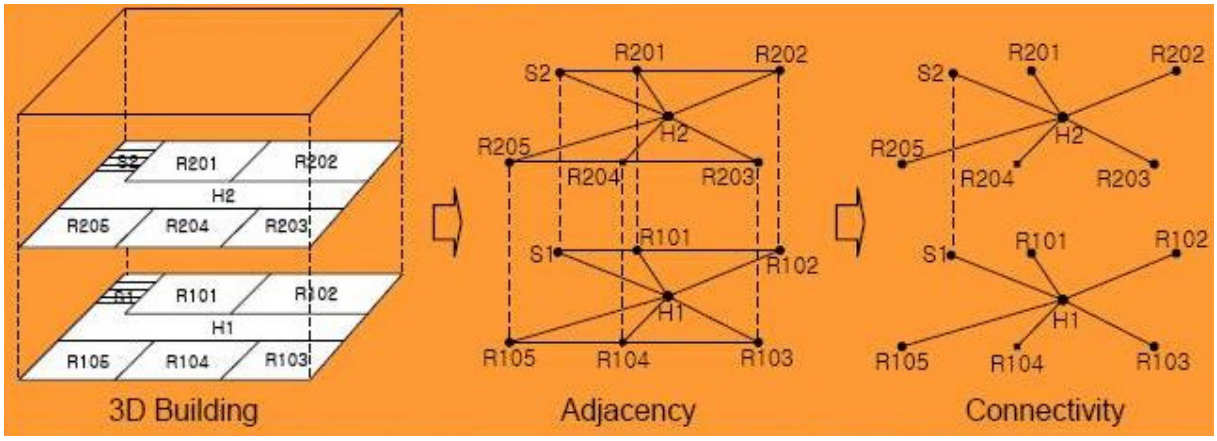


Figure 10 Overview to create a 3D connectivity network (Stevens, Choi 2006)**.**

An effective and well structured model of the 3D scene space is needed to support walking routing. In research for emergency response in urban areas, evacuation and routing is studied (Meijers, Zlatanova et al. 2005)(Choi, Lee 2008)(Lee 2004). Navigation inside and outside buildings is of vital importance in emergency situations and therefore various 3D data models are studied. These 3D models are categorized in four categories: 3D geometric models, 3D topological and graph models, 3D city models and 3D CAD models. The routing algorithms are working on networks. In this 'emergency response' research it is found that the 3D topological and graph models are best suitable for 3D routing. Construction of these topological and graph models is described in literature (Lee 2004) (Stevens, Choi 2006). Figure 10 gives a visual representation creating a 3D connectivity graph from a 3D building model. From the 3D building model all adjacent spaces (rooms, halls and stair) from each space are defined. In this figure e.g. room R104 is adjacent to the hall H1, room R106 (left), room R103 (right) and room R204 (above). Then the door and pass ways are evaluated and the result is a 3D connectivity graph. The floor plan with the location of the lower floor is shown in the middle in Figure 11with this connectivity graph a list can be calculated showing only which spaces are passed when walking from origin to destination. This 3D connectivity graph is input for a process to create a walk routing network in 3D, called geometric network, in this model as shown in Figure 11.

To construct this walk routing network the connectivity graph is fitted on the floor plans and walking routes in the spaces are determined for the spaces (Medial axis transformation MAT (Lee 2004). The walking routes in the spaces are defined by adding one or more nodes to each space and connecting these nodes. In this way the logical walking routes in the spaces (called "hidden paths" in this thesis) are modelled. These walking routes in and the connection routes between, the spaces are the ingredients of the 3D geometric network.

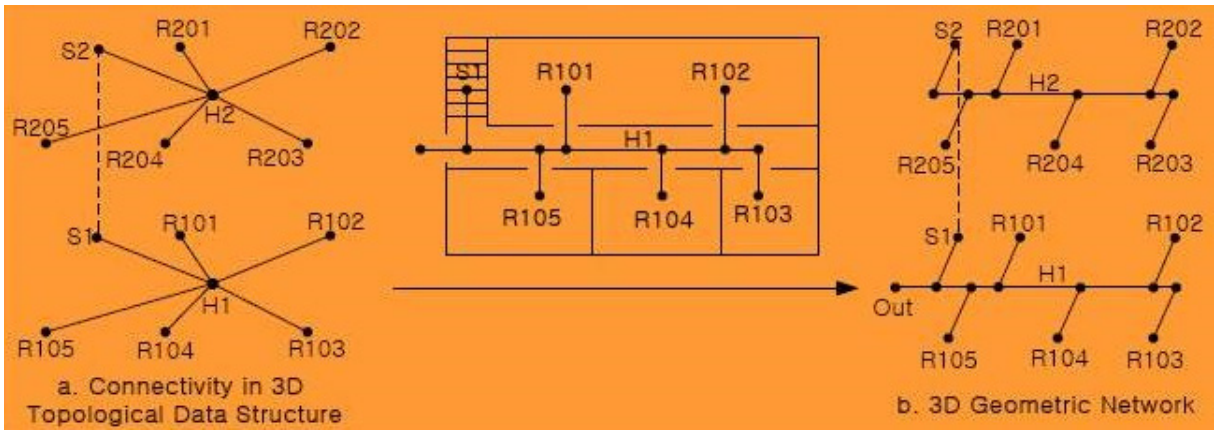


Figure 11 Overview to create a 3D Geometric Network (Stevens, Choi 2006)**.**

3.6.2 Navigable spaces model

In the model of Slingby (Slingsby, Raper 2008) pedestrian space combines geometrical, semantic and navigational aspects. In this model 3D spaces are represented only by their lower surfaces. These lower surfaces are used to model pedestrian navigation in 3D. For the objects describe by the lower surfaces, existence, accessibility, pedestrian direction and construction material can be described using attributes. Some of these attributes are time depended others are pedestrian dependent. Its semantic model consist of 'barriers' (walls), 'portals' (doors and windows), 'teleports' (lifts) and 'spaces' (specific delineations of space). Persistency information is provided through lists of unique or recurring time periods (e.g. some barriers only exist at certain times of day). Access granting information is provided as lists of the times (unique or recurring) at which access is granted and (optionally) a specific or specific type of pedestrian. In this context, a pedestrian has a number of characteristics (e.g. age, gender) and may be in procession of one or more door keys or access cards. Direction of passage information and construction material information can be applied to some of these objects. Barriers and openings are classified according to the ease of unauthorised access, e.g. how easily it can be passed with or without damage. Pedestrians have a maximum ease threshold of barriers for which they would be willing to breach. This might be context-dependent, for example if there is an emergency. Using this information, the model attempts to incorporate some of the micro scale details of pedestrian access. As stated, a pedestrian has attributes, may have a collection of access cards of door keys has a threshold amount for gaining unauthorised access. Additionally, a pedestrian has a step-height he or she is able to negotiate, which would be zero or very small for a wheelchair user. Although this model gives a good basis for a pedestrian model it semantics containing 'barriers' (walls), 'portals' (doors and windows),'teleports' (lifts) and 'spaces' (specific delineations of space) is not in line with semantics used in the field of public transport.

4 TOWARDS A NEW MODEL FOR PEDESTRIAN ROUTING IN 3D SPACE

Interesting are the differences in the models found in literature. All models are developed with a certain purpose in mind. Some models are focusing on indoor pedestrian navigation using networks, others models are using the pedestrian surfaces and focusing on possible barriers in pedestrian routes. Urban models are focusing on outdoor 3D and building models on indoor. Only the Stop Place Model (IFOPT 2007) has semantics for the public transport scene space, but is not focusing on 3D geographical aspects. A new model must integrate specific aspects of different models, connecting outdoor and indoor pedestrian routing to public transport.

To come to a new model as research approach the requirements for an effective and well structured geo-data model were defined after the scope definition in this research. These requirements are part of the prototype approach and used to develop a geo-data model. A separation is made between functional requirements and non function requirements. The functional requirements are based on a general use case for pre-trip journey planning for the pedestrian parts of a journey. The non functional requirements are based on the scope definition and agreements made at the start of the research. This chapter starts with describing these functional and non functional requirements. After this description of requirements, a description of this general use case and the scenario for evaluation of a new geo-data model is given.

4.1 Functional requirements

In this research the walk part of a transport journey is seen as integrated part of a public transport advice. The walk part of a public transport journey exists in 3 parts of a journey (Rüetschi, Timpf 2005).:

1) When accessing the system (access)
2) During transfer (interchange)
3) leaving the system (egress)

In Figure 2 the interaction was showed between these parts of a journey. The division of a public transport trip is used as model for specifying the requirements.

To come to the functional requirements for route choice the aspects which influences an optimal route choice for an individual traveller where studied. In literature (Lee 2005) it is described that on-trip pedestrian route choice is based on travel time and time pressure, physical effort safety and conform personnel characteristics and vicinity. In appendix D is described how these aspects are related to on-trip optimal pedestrian route choice. This research is limited to pre-trip pedestrian route selection, taking only pedestrian walk duration, and access and mobility restrictions, into account as functional requirements for selecting an optimal pedestrian route.

The following functional requirements were defined:

1) The geo-data model must support access and egress of public transport locations.
2) The geo-data model needs to connect walking routes to the public network model used by public transport planners.
3) Detailed modelling of pedestrian areas inside public transport location must be supported
4) The geo-data model must be able to support shortest path routing methods based on walk times taking individual travellers access or mobility restrictions into account.
5) The geo-data model must be able to support visualization of walking routes in 3D.

4.2 Non functional requirements

To evaluate the effectiveness of the geo-data model a prototype will be implemented in Oracle Spatial 11. To well structure the geo-data model following the guidelines will be taken into consideration:

1) It must be possible to implement the geo-data model in a spatial database.
2) An object approach in the design of the walk able areas and 3D elements.
3) A semantic model in line with public transport semantics.
4) Applying where possible (Open or industry) standards in design and implementation

4.3 Use case for pretrip journey planning to prepare for way finding

In this research one general use case for pre-trip planning of the walk parts of a public transport journey is defined. Figure 12 shows this general used case in a UML use case diagram. In this figure is a (general) traveller is seen as actor. This actor is representing all kind of travellers preparing their journeys including:

- Travellers with mobility restrictions (e.g. wheelchair users)
- Travellers without mobility restrictions
- Travellers with auxiliary needs (e.g. travellers with shopping needs)
- Travellers with individual needs (e.g. tourists needing extra transfer time to catch a international train)

In Figure 12 this general use case of preparing for the walk parts of a public transport journey is divided in several sub use cases. These sub used cases are categorized to logical parts of a public transport advice by colour:

- Green: use cases related to a public transport journey including the start and origin of a journey and access and egress of public transport vehicles in a journey
- Blue: use cases related to entrance and exits of public transport locations
- Yellow: use cases related to pedestrian route choice inside a public transport location.

The relation of the sub use cases to access, interchange and egress nodes of public transport (see in Figure 2) is showed by the 3 bounding boxes.

To cover the objective of this research to support access, interchange and egress walk-routing in the context of pre-trip route planning and as addition to a public transport journey advice from a public transport route planner the following use cases with are defined:

- 1. Locating the entrance (s) of a public transport place in relation to the surroundings.
- 2. Detailed routing between entrance and access of public transport.
- 3. Detailed routing for inside a public transport place including detailed walk time.
- 4. Detailed routing between egress of public transport and exit of public transport place
- 5. Locating the exit (s) in a public transport place

The two use case are directly related to support walking time and mobility restrictions and defined as necessary parts of use case 3:

- 3.1 Determine connection time for all walk routes
- 3.2 Support routing for travellers in access or mobility restrictions.

The individual traveller needs are influencing all use cases. Support for auxiliary services (e.g. shopping, buying a ticket) is seen as extensions (extra) to the standard use case 3. The use cases 1.1, 2.1, 4.1 and 5.2 are the use cases related to public transport travel advises and could be fulfilled with public transport (address to address) planners supporting the individual traveller. The use cases "1.2

selecting an entrance" and "5.1 selecting an exit" are also strong related to the travellers needs (e.g. wheelchair users could use specific entrances or exits based on their specific mobility constraints).

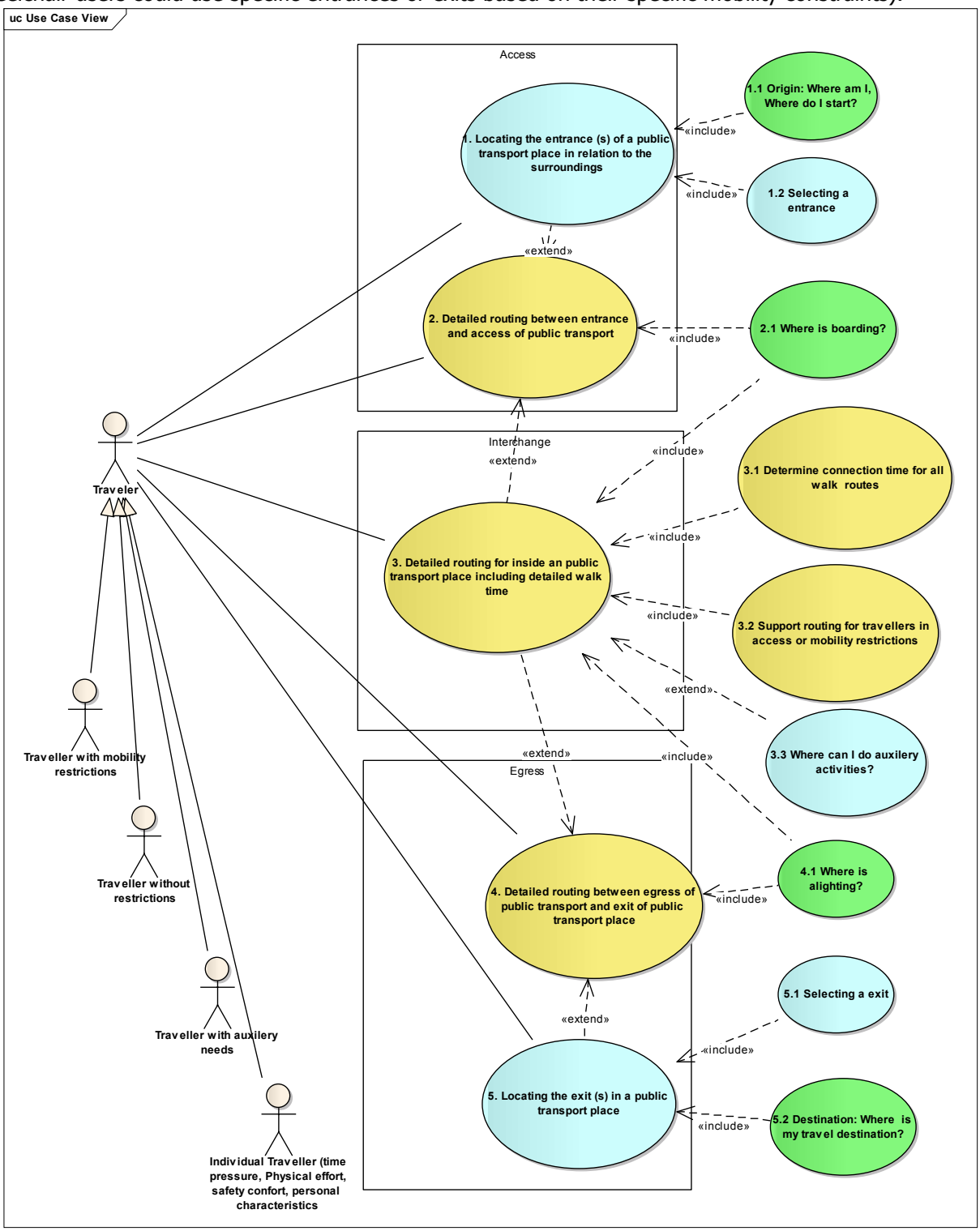


Figure 12 UML Use cases diagram for a public transport pedestrian pre-trip planning

4.4 Scenes and scenarios to validate with a new model

At the start of this research possible public transport journeys where studied between main entrances of largest building of the GIMA 4 course locations in Delft, Enschede, Utrecht en Wageningen. A journey between the Forum building in Wageningen and Utrecht University library, via bus stop "Droevendaalse steeg (direction Ede)", train station Ede-Wageningen and Utrecht Central station, was selected because it is a multimodal journey and the first three public transport locations passed in this journey are of increasing complexity. From this journey the first 3 public transport places where candidate to implemented in the prototype of this research. At first a simple public transport place consisting of two bus stops at each side of the road to model access and egress for the modality bus in 2D. Second a location containing one bus stop and a train station to focus modelling on multimodality and 3D accessibility (including underground pedestrian passageway, ramps, stairs, elevator, pavement and platforms). Third, a complex public transport place of national importance containing a complex train station, a complex city bus station, a regional bus station and a local tram stop. The size and complexity of this third location made construction and data collection to evaluate scenarios with this location unfortunately not possible in the timeframe of this research.

With the first simple public transport place the following scenario's focusing on unrestricted routing with a fixed speed were validated:

1) Access a Bus stop from a fixed location
2) Egress a Bus stop to a fixed location
3) Navigate "inside" a bus stop.

The following scenarios for restricted and unrestricted travellers with 3 different speed types where evaluated with the model of the second location focusing on multimodality and 3D accessibility:

4) Access a simple multi modal transport place
5) Egress a multi modal transport place
6) Navigate "inside" a multi modal transport place, considering steps, stairs, ramps and elevators
7) Change modality bus to train
8) Change modality train to bus

The scenarios related to the third location, a complex multimodal transport place of national importance, where unfortunately not evaluated:

1) Access a complex multi modal transport place
2) Egress a complex multi modal transport place
3) Navigate "inside" a complex multi modal transport place, considering step, stairs, ramps, elevators and escalators
4) Object ticket machines
5) Change modality using tram

5 A NEW CONCEPTUAL MODEL FOR PEDESTRIAN ROUTING

In this research a new conceptual model to support access, interchange and egress walk-routing in multimodal travel advices was developed. This chapter describes this model in detail. The Unified Modelling Language (UML) is used as formal description of the geo data model in this conceptual model. This chapter describes this conceptual model in detail starting with a functional overview of the model. After this functional overview the details and relations of the objects in this model are described. This is done in the following way:

- **Focusing on a single space:** The semantic and geometrical details of a space where walking is possible is modelled with a pedestrian accessible surface called "NodeSurface" in this model in paragraph 5.2.

- **Focusing on the connection of spaces**: Spaces modelled with pedestrian accessible surfaces are connected to each other to create a pedestrian network to support pedestrian routing in scene space. These connections are called LinkSurfaces (paragraph 5.3)

- **Focusing on walk routes in space**: Modelling walking routes can be done in different ways. The model support the concept of modelling logical walking routes in spaces (called "hidden paths" in this model) and the concept of ignoring the modelling of logical walk routes in spaces (called "walk around" in this model). These concepts are described in paragraph 5.4.

- **Focusing on modelling and visualization of level changes:** Entrances, stairs, ramps and elevators are relevant in scene space in detail is described how the different visual elements are modelled. (Paragraph 5.5)

- **Focusing on travellers and their needs:** How traveller preferences are model to support pedestrian routing including mobility needs and walk speed. (paragraph 5.7.1 and 5.7.2)

- **Focusing on finding optimal routes:** In detail is describe how the pedestrian network is modelled to obtain a optimal pedestrian route an and how the pedestrian route is connected to public transport (paragraph 5.6 and 5.8)

5.1 Overview of the conceptual model.

The conceptual model is organized around 3 functionalities:

1. Public transport
2. Pedestrian
3. Visualization

This organization of functionalities is derived by the author from analyses of models found in literature and analysing the use case for pre trip journey planning to prepare for way finding as described in the previous chapter. The separation of public transport and pedestrian encompasses the separation of (public transport) network space and scene space as described in paragraph 3.2. Similar concepts of the pedestrian functionality where found in models supporting pedestrian routing found in literature. The visualization encompasses unfolding the meaning of objects relevant for pedestrian navigation, combining visual and textual information, using the semantics derived from the IFOPT standard in travel advises. In UML organization of objects (classes) is done in packages, Figure 13 shows this organization of objects (classes) according to the functionalities. In the UML diagrams colours are used to focus on this organization of functionalities in packages. Green is used for the public transport functionality package, Yellow for pedestrian functionality package and Blue for the visualisation functionality package.

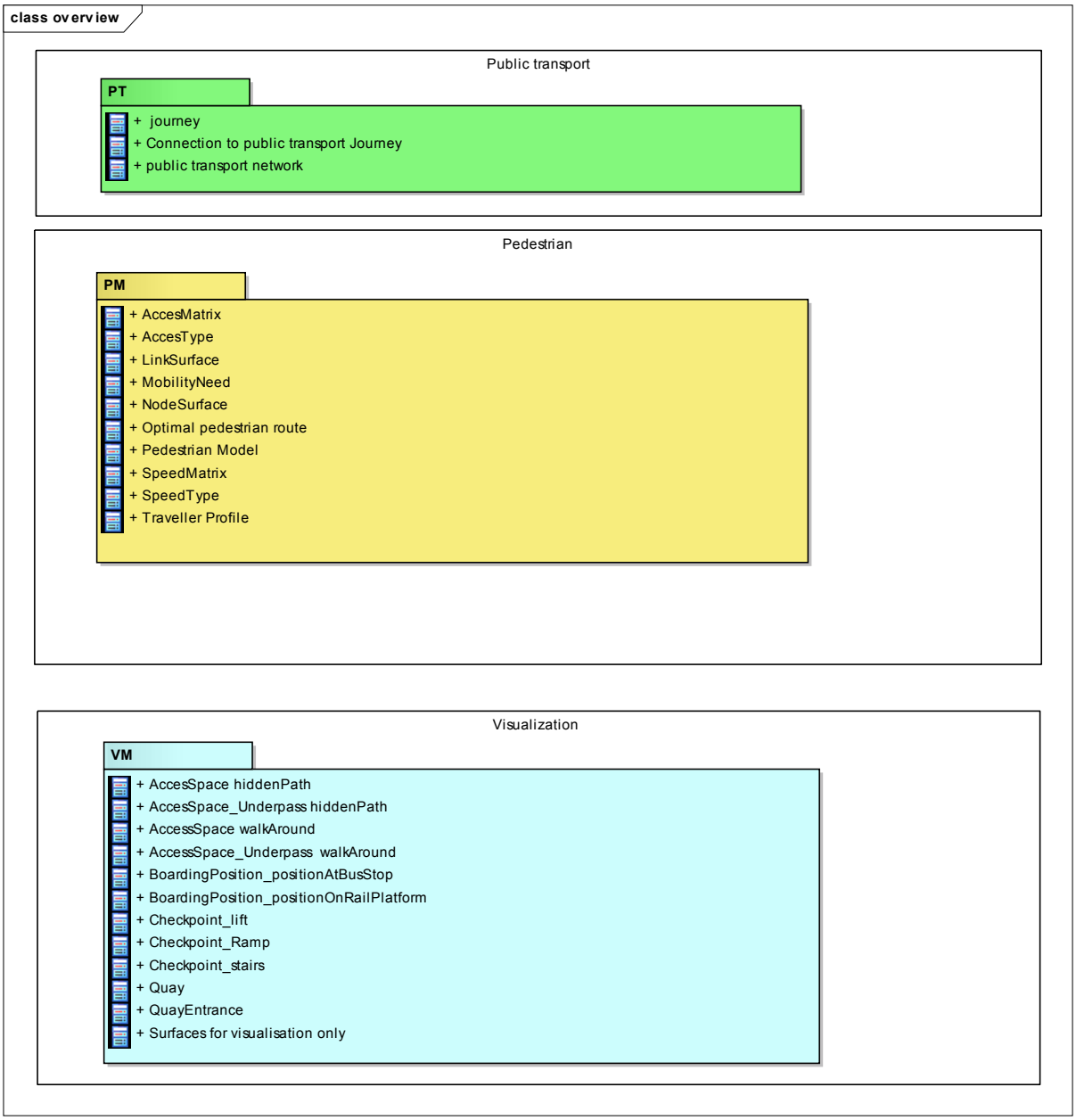


Figure 13 UML Overview functional domains of the conceptual model for pedestrian routing

The pedestrian functionality contains to basic functionalities, to store the geometry for pedestrian scene space and to find optimal routes in this scene space. The pedestrian package uses the Simple Feature Implementation Specification for SQL (OGC 1999) as standard to store spatial data. The data types surface and multisurface as described in paragraph 3.4.1 are used. To focus on this dependency a package for this standard named "Simple Feature" is added to the conceptual model. Furthermore a network, consisting of node and links is needed to model connected scene spaces and is used to support finding an optimal pedestrian route. This network functionality is enclosed in a package "Network". In Figure 14 these two packages are shown as base packages, the network package in purple and the simple feature package in red. Figure 14 is also showing the content of the 3 other classes and relations. All these classes and relations are described in detail in de remainder of this chapter.

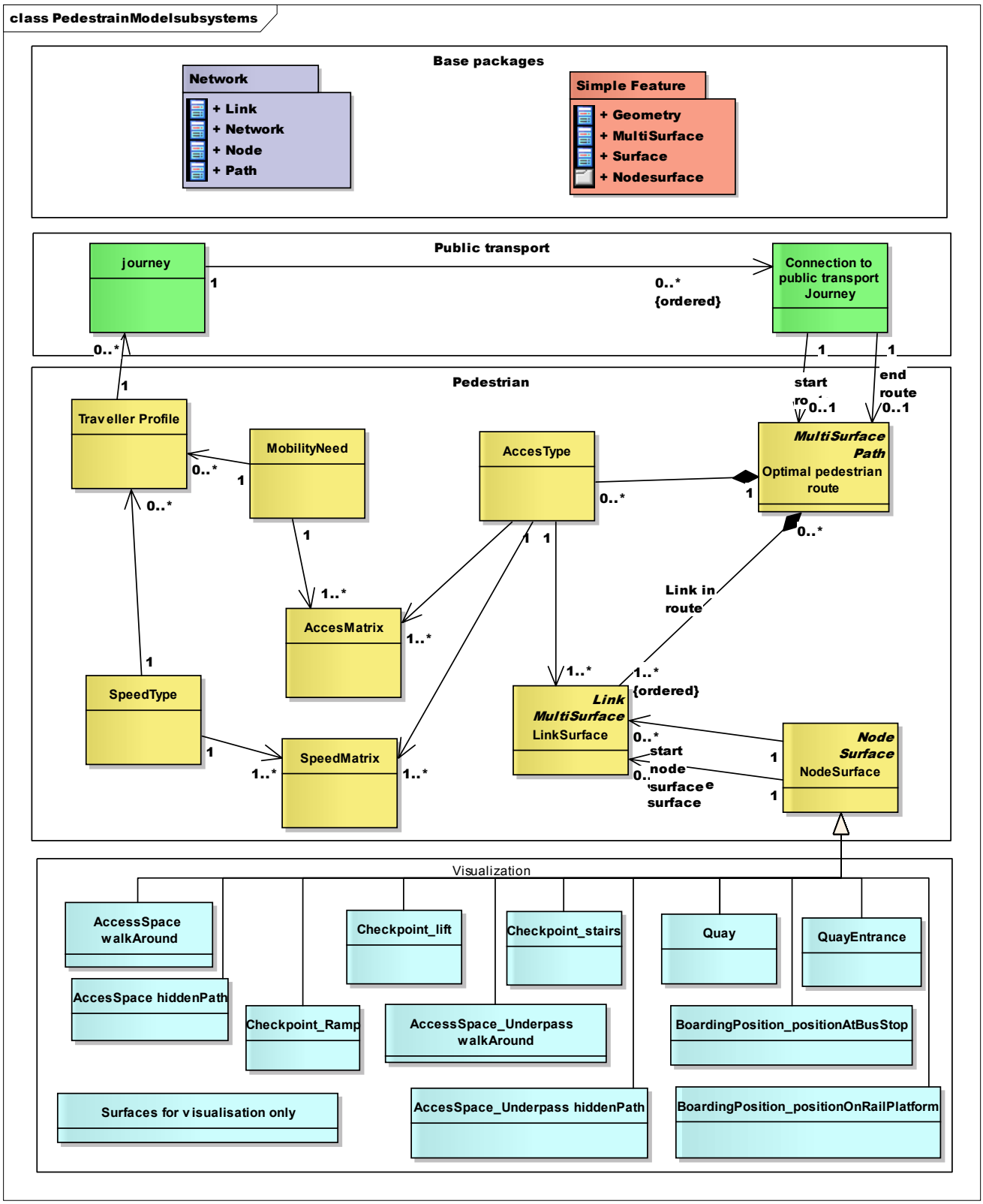


Figure 14 UML Overview class diagram of the conceptual model for pedestrian routing

5.2 NodeSurface a pedestrian accessible surface.

The conceptual model focuses, similar as the model of Slingby (Slingsby, Raper 2008), on the lower surfaces of accessible 3D spaces. 3D pedestrian accessible spaces are represented only by their lower surfaces. To align this model to the IFOPT Stop Place Model these lower surfaces are considered semantically identical as the accessible place in IFOPT. The IFOPT standard focuses on describing the semantics for this accessible place. As described before IFOPT has a projection to a GIS model. This conceptual model differs because it integrates semantic and 3D geographical representation in one model. For it semantics it is using the IFOPT names and to represent the objects it uses the Simple Feature Implementation Specification for SQL. For outside this lower surface of accessible 3D spaces is slightly comparable with a *TrafficArea In* CityGML objects. For indoor it is comparable with a CityGML Floor Surface. A difference is that in CityGML basis for these objects is the GML:MultiSurface where in the pedestrian model the basis for an accessible surface is a Simple Feature Surface (In Figure 16 classes from the simple feature package are showed in Red). Appendix C of this thesis contains a UML diagram with an implementation of CityGML surfaces in a spatial database (Kolbe, König et al. 2009). This difference can be seen comparing this figure with Figure 16 showing the implementation of the pedestrian accessible surface called NodeSurface in this model.

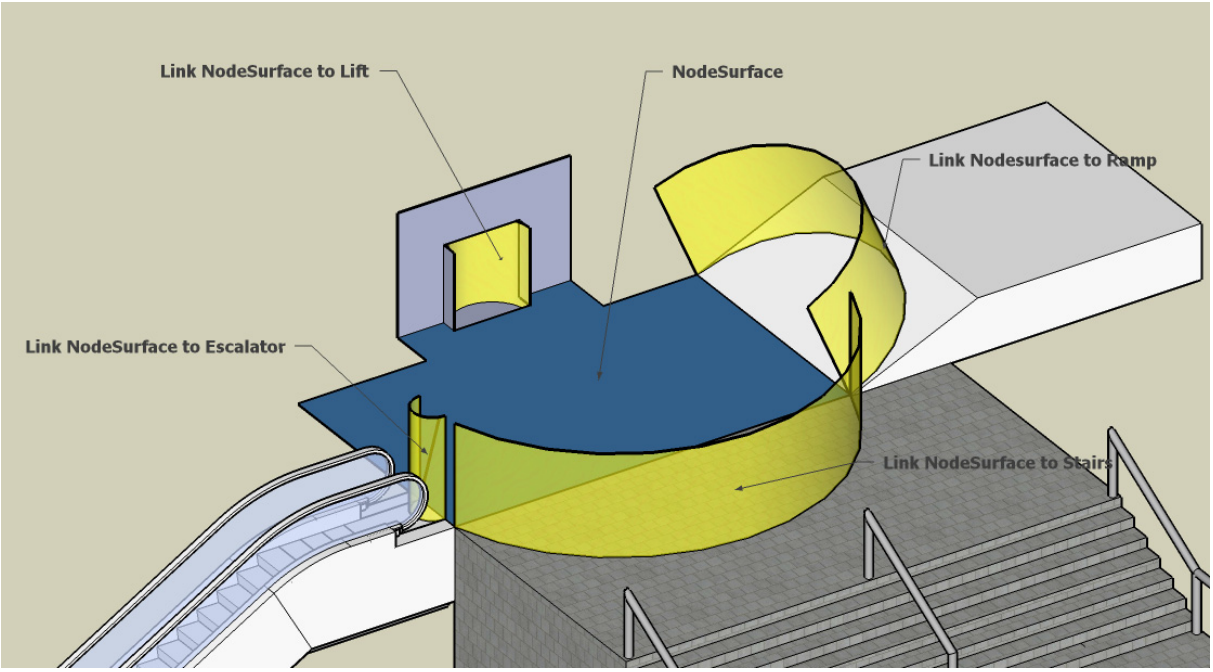


Figure 15 Basic elements of the pedestrian model

In Figure 15 a pedestrian accessible surface is shown in blue. This imaginary surface (NodeSurface) is surrounded by other accessible surfaces. These surfaces are the other basic types of accessible surfaces, an escalator, a lift a ramp and stairs. The selected NodeSurface (in blue) is a perfect flat surface and the surfaces itself has no access restrictions.
To support traveller routing the following questions are relevant for the blue selected NodeSurface in a pedestrian route:

- Is this blue surface accessible?
- Has the blue surface access limitations for specific traveller groups?
- Is the accessibility of the surface limited in time?
- How to reach this surface? (entrances)
- What is accessible from this surface? (exits)
- With access routes are possible on this surface?

Important is that answers to these questions are depending on adjacent spaces (Entrance and exit) of the blue NodeSurface. To visualize this relation to adjacent spaces in Figure 15, yellow views "Curtains" are drawn on the adjacent surfaces of the blue NodeSurface. These yellow views can be seen as what an imaginary traveller sees standing on the blue NodeSurface of adjacent spaces.[4] These yellow views "Curtains" are draw only to explain the model; they are showing the possible entrances a traveller can take to move to other NodeSurface. These yellow views can be seen as pictures taken from the blue NodeSurface to describe possible directions. The fact that the escalator is working and the direction of it is not shown in the image. For a traveller and thus the pedestrian model this is actually important information, because it influences route choice.

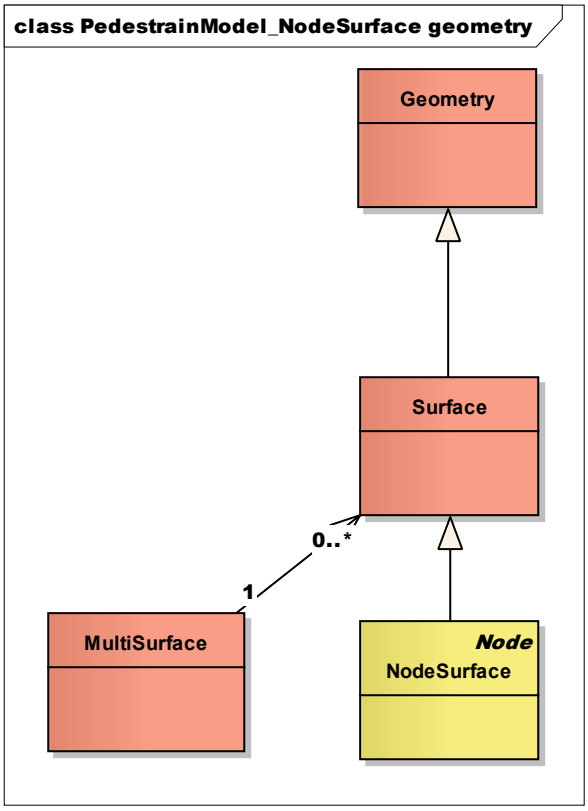


Figure 16 Simple Feature Surface as basis for a Pedestrian Accessible surface (NodeSurface)

[4] To simplify the figure walls restricting the NodeSurface between escalator lift and ramp are not drawn, the imaginary traveller is not supposed to jump of the surface between escalator and lift.

5.3 LinkSurface a pedestrian accessible surface with a direction

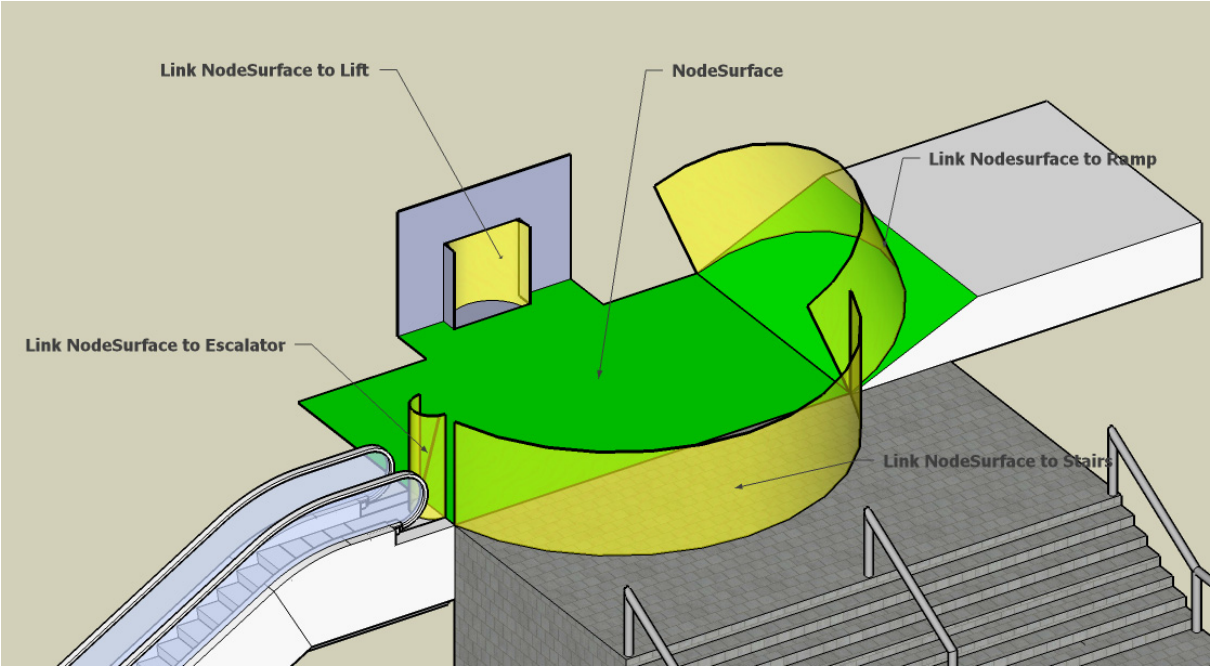


Figure 17 MultiSurface Link from NodeSurface to Ramp in green

As describe in the previous paragraph entrance and exit of adjacent spaces are important for routing choice. In CityGML LoD3, the openings in BoundarySurface objects (doors and windows) can be represented as thematic objects. In the pedestrian model the logical links (shown in the image in yellow) between the blue surface and adjacent spaces are modelled as the combined surface of the origin (Start_nodeSurface) surface and the destination (End_nodeSurface). The link from the NodeSurface to Ramp (Surface) is shown in Figure 17. A link, called LinkSurface in this model, is a (derived) multiSurface which can also be seen in Figure 19. Order of the surfaces is important in his multiSurface because the order defines the direction of the link. The model restricts this multiSurface to a surface consisting only two surfaces with only one connecting border. No disconnected surfaces are allowed in this multiSurface. This is done because route choice in this model is based on selecting relevant directions to unique entrances and exits of NodeSurfaces. Figure 18 is an example of an invalid link (surface). The link is invalid because two connecting boarders are between the NodeSurface described in the previous paragraph and the ramp. In Figure 18 a relevant difference exits in taking the left or the right entrance to the ramp. This difference is relevant in Figure 18 because in this case an imaginary traveller does not know that both entrances are leading to one single space (the ramp). To correct this situation, the node surface or the ramp must be split or walk ways (hidden paths described in the next paragraph) must be modelled. No holes are allowed in connecting boarders of NodeSurfaces. Holes are also not allowed in the NodeSurfaces. Small holes, not relevant for route choice, e.g. pillars in underpass, are not modelled in NodeSurfaces. Pillars are modelled as elements for visualisation only in this model. Furthermore NodeSurfaces interior to other NodeSurfaces are not allowed in this MultiSurface. Selecting optimal directions (a route choice) from an interior NodeSurface (island) is determined by other (destination) NodeSurfaces not by the first adjacent NodeSurface.

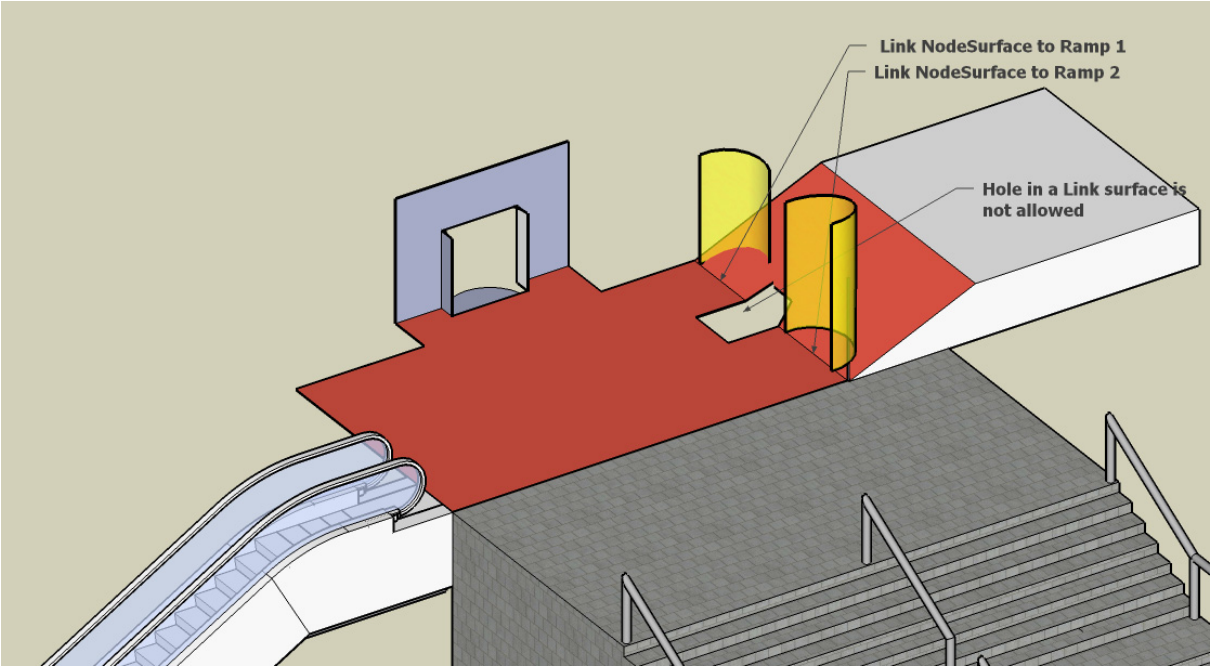


Figure 18 Invalid MultiSurface Link from NodeSurface to Ramp in Red

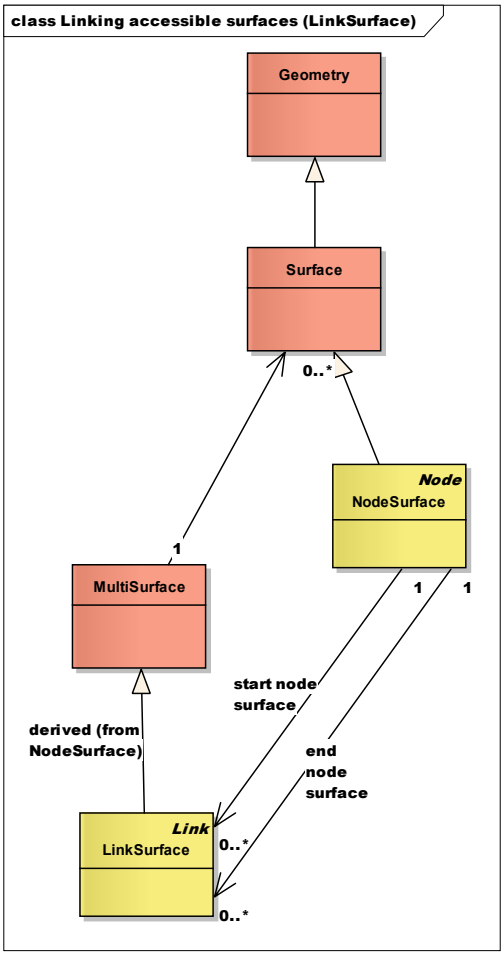


Figure 19 Class diagram showing LinkSurface as multiSurface and the start and end node roles

5.4 Modelling NodeSurfaces; "hidden paths" or "walking around areas"

When modelling pedestrian paths the model support two ways of drawing pedestrian access paths. In Figure 17 the route a traveller follows when exiting the lift and walking to the ramp is not defined. The model supports the concept that the traveller walks freely to the ramp. This concept of free walking makes it possible to use a model only consisting of access areas and to model a pedestrian walk in a global way. For applications where more detail is needed the model also supports the concept of modelling logical walking routes in the spaces, called "hidden paths" in this model. (Other approaches to model this detailed logical walking route in spaces, e.g. models used for movement planning in cross country, are interesting but considered outside of the scope of this research). When modelling hidden paths the NodeSurfaces as defined in the previous paragraph are split in more detailed NodeSurfaces. Both ways of modelling walk can be used in one model. When an algorithm calculates the optimal path it can decide to use a hidden path NodeSurface or a "walk around" NodeSurface as part of an optimal path based on the attributes of these NodeSurfaces. In general the hidden paths will have a shorter walking distance then the "walk around" surfaces and the hidden path will be in favour. But when some activity must de done on a surface e.g. "buying a ticket at a vending machine" and no hidden paths are to the place where this activity occurs then the 'Walk around surface" can be used by the route calculation algorithm to reach this place. Furthermore when modelling the hidden path the original surface can also be seen as an area divided in 3 surfaces. A surface connecting the escalator to the hidden path(gray) at the right hand when leaving the elevator, the hidden path itself (green) and the corner area (gray) at the left hand when leaving the elevator. When modelling hidden paths on could decide to leave this left hand and right hand surfaces out of a pedestrian model (deciding that the gray area's are not a NodeSurface and use as a surfaces for visualisation only). The gray area is than inaccessible for the model, this forces the use of the hidden path between the elevator and the ramp and makes the escalator inaccessible. As seen in Figure 21 these inaccessible surfaces are not included in the pedestrian model, they are only available for visualisation. Important in Figure 21 is the fact that no relation between surfaces for visualisation only and the collection of NodeSurfaces exists. In the model hidden paths and the original walk around area are only named different, the hierarchical relation of a hidden path to its original walk around area is not modelled. In this way this conceptual model differs from the IFOPT Stop Place Model, where the hierarchical relations can be modelled. In IFOPT an AccessSpace can be modelled as subzone of another AccessSpace (The AccessSpace Class and the subzone relation can be found in the UML Class diagram in Appendix B).

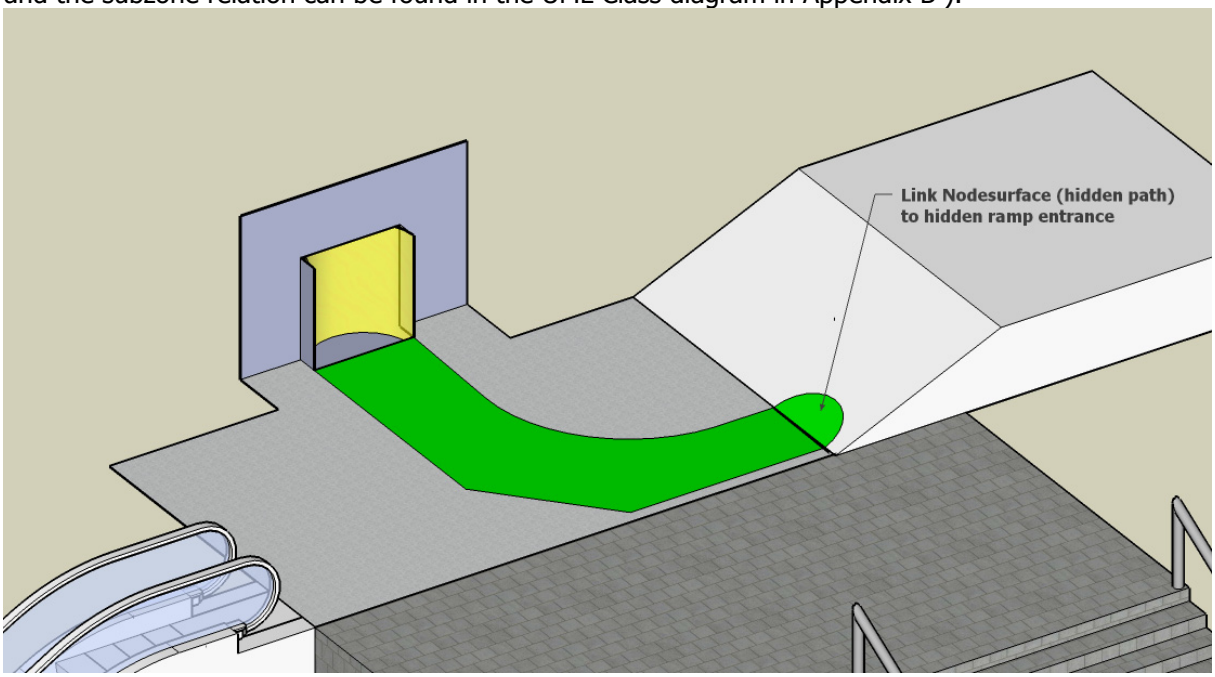


Figure 20 Hidden path and hidden entrance as link in green

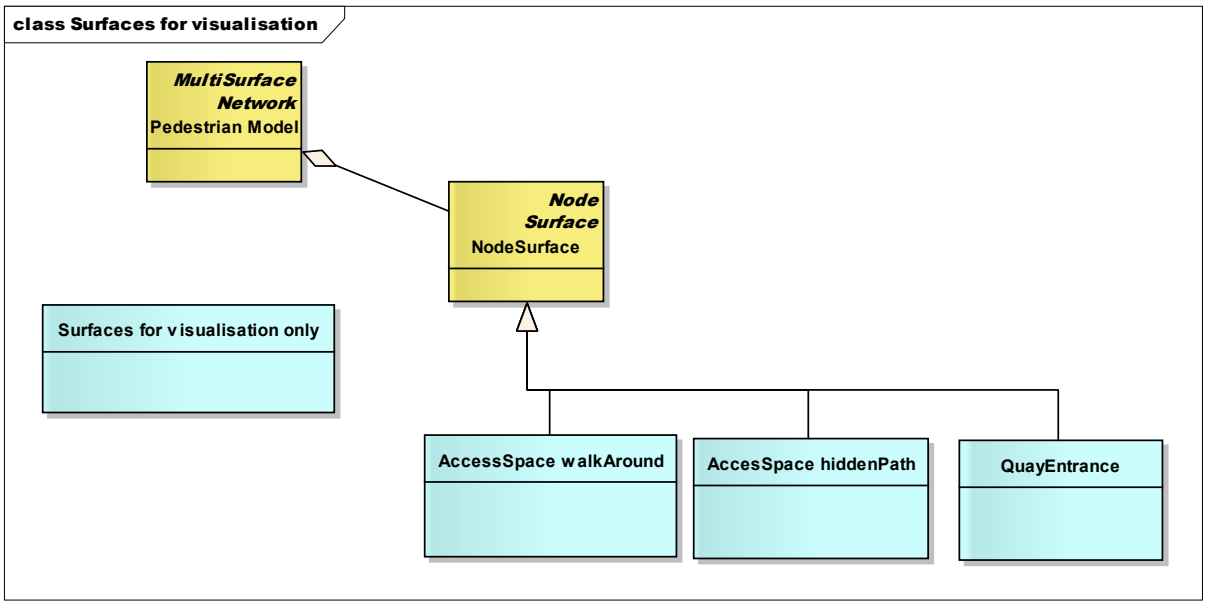


Figure 21 Class diagram Surfaces for visualisation and walk modelling

5.5 Modelling entrances, ramps stairs, escalators and lifts with NodeSurfaces

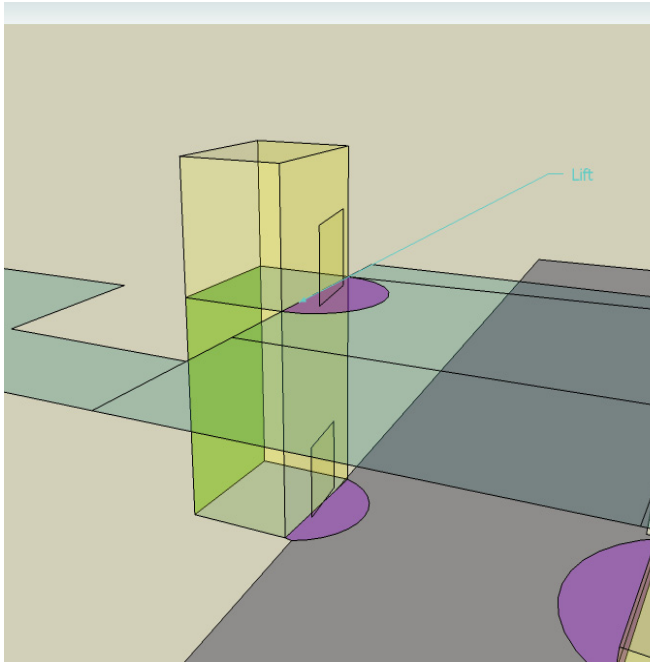


Figure 22 Lift in Google SketchUp (with visualisation layer)

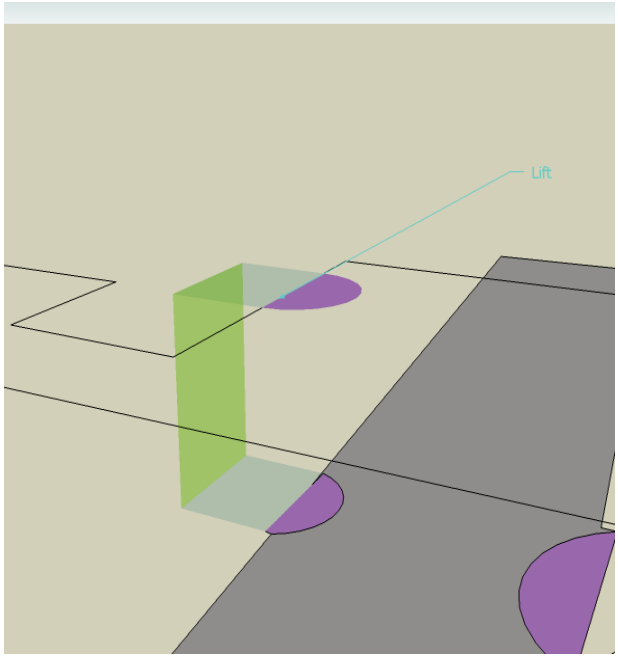


Figure 23 Lift in Pedestrian model (without visualisation layer)

All ramps, stairs lifts are modelled as objects deriving attributes and behaviour from NodeSurfaces, in fact in the conceptual model these objects are (more specific) type NodeSurface. To visualize these types in 3D drawings different colours were used for different types. Figure 31 on page 52 illustrates this colouring; each type of NodeSurface is drawn on a specific layer in a Google SketchUp drawing. For visualisation in SketchUp models can be placed as an extra visualisation layer on top of the model as shown in Figure 31 on page 52. The surfaces of these SketchUp models for visualisation are "Surfaces for visualisation only" in the conceptual model. For the naming of the different NodeSurface

object types IFOPT names was used. In IFOPT the term "checkpoint" is used for an object in a space that slowdown pedestrian speed (In appendix B the IFOPT Checkpoint class links to the stop place hierarchy). In Figure 24 this modelling of NodeSurface types with IFOPT names is showed in a UML Class diagram.

The ramps and stairs are modelled as steep NodeSurfaces connecting the different levels. Treads and risers are surfaces for visualisation only and not related to the stair (Checkpoint_Stairs) in pedestrian model. Direction is modelled by using a Checkpoint stairs NodeSurface and an adjacent NodeSurface using a LinkSurface. Stairs (up) from an underpass to a higher train platform (IFOPT Quay) can be modelled with 3 NodeSurfaces and two LinkSurfaces. One LinkSurface have a NodeSurface for the underpass (Accesspace_Underpass walkAround) as start_NodeSurface and the stairs (Checkpoint_Stairs) as end_NodeSurface plus another LinkSurface with the stairs (Checkpoint_Stairs) as start_NodeSurface and the platform (Quay) as end_NodeSurface. Hidden paths, Logical walk ways, used as entrance to other space are called "Hidden" entrances. These "Hidden entrances" to platforms are modelled using an IFOPT name QuayEntrance.

Figure 22 shows how lifts are modelled in the pedestrian model. In Figure 23 the faces relevant for the lift in the pedestrian model are shown (e.g. the "green" platform level is transparent in this image). In purple the entrances to the lift at platform level and the lift entrance position in the underpass. Important is a surface the "back wall" of the lift (in green) connecting the two floors with each other. This back wall is an exceptional NodeSurface because is not a lower surface of the lift space and it is not a pedestrian accessible area. For this exceptional NodeSurface also special calculations methods for "walk"- duration are developed and described in chapter 6.1.3. This "back wall" is the basis of the connection between floors and modelled as NodeSurface (Checkpoint_lift). This "back wall" is used as connecting NodeSurface because in the lift (on the lift floor) walking around is possible. If e.g. the "front wall" was used, the inside of the lift (AccessSpace) was not part in an optimal connection between the levels. The lift in the studied location only connects two flours the underpass and the platform, therefore the "back wall consist only one vertical surface. The purple surfaces can be mapped to the QuayEntrance. A small overhead is taken in horizontal distance because the "ground floor" (AccesSpace) of the inside space of the lift is modelled twice, one time as floor on underpass level and one time as floor on platform level. In reality only one moving lift floor exists.

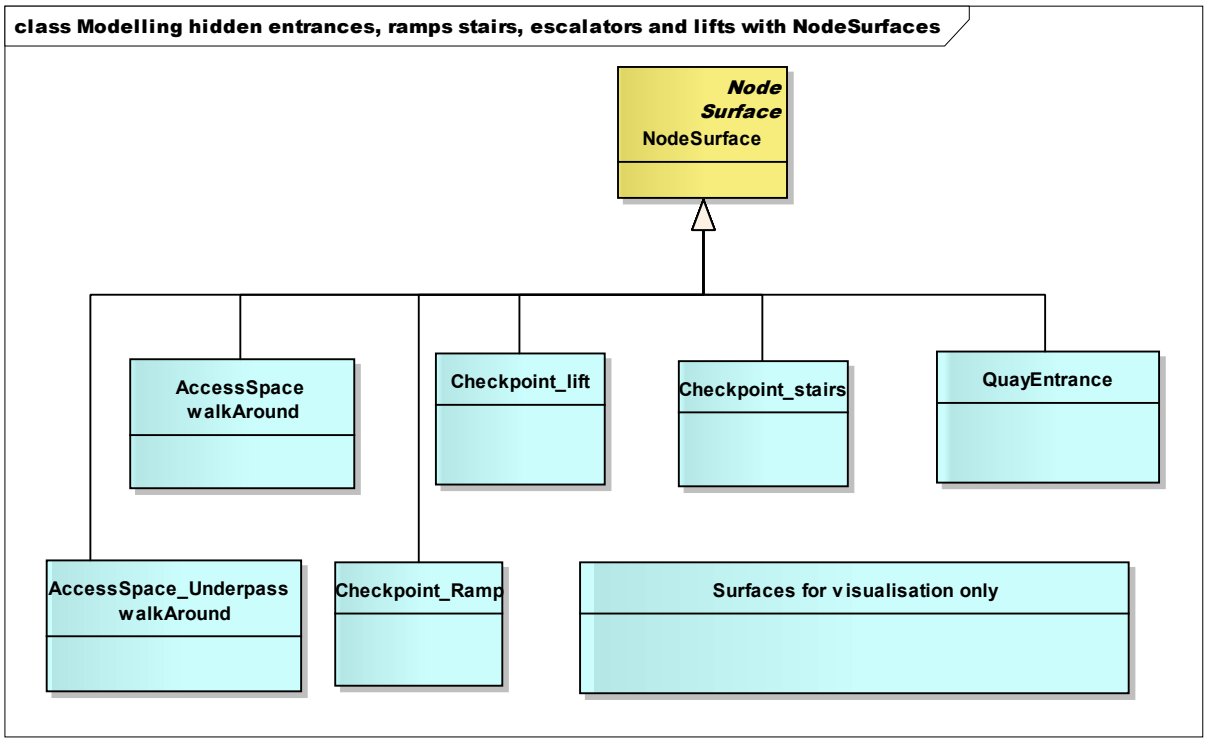


Figure 24 Class diagram hidden entrances, ramps stairs, escalators and lifts are NodeSurfaces with IFOPT naming

The modelling in the visualisation package of NodeSurfaces of different type differs from modelling in IFOPT stop place model. In IFOPT Stop Place Model these locations are in an inherence tree to define the relevant locations for travellers. It is used to define a stop place on a global level or to define parts of a stop place e.g. the booking hall (an Access space), a platform (quay), the place where the vehicles are stopping (vehicle stopping place). In appendix B the UML Class diagram of the Stop Place Model can be found. This partition of stop places is not done in this model, because it is already exists in the IFOPT standard. The semantics used in this model are directly related to the IFOPT semantics and therefore this hierarchy can be directly added to the classes with IFOPT related names in the visualisation package.

5.6 Pedestrian Network and Optimal pedestrian route

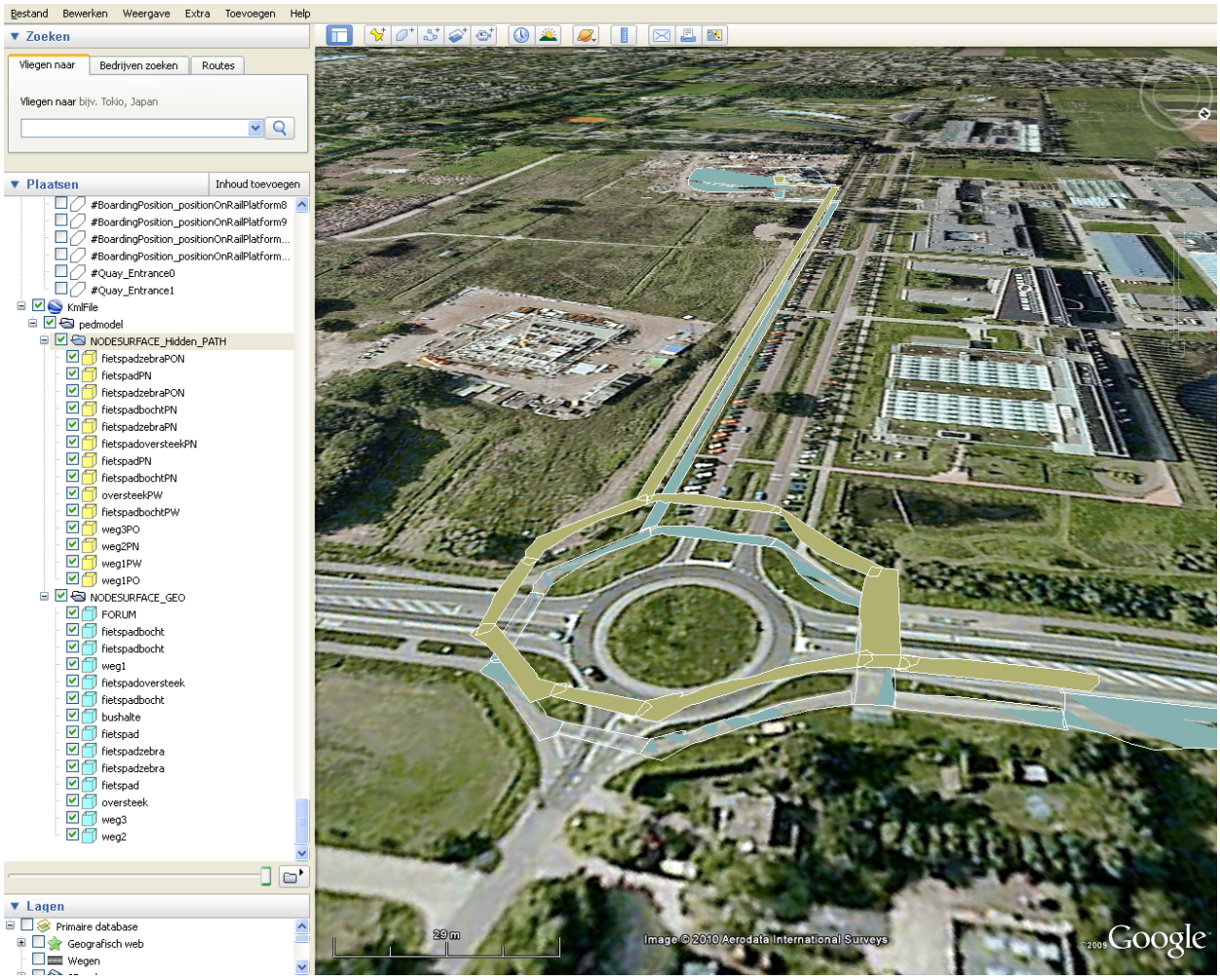


Figure 25 Example surfaces in the Google Earth to test 2D routing

In the pedestrian model a network model is used. This network model, consisting of node and links is needed to model connected scene spaces and is used to support finding an optimal pedestrian route. This basic network functionality is enclosed the package "Network". All connecting scene spaces in this pedestrian model are modelled as a network where the surfaces are the nodes (NodeSurfaces) connected by the links (LinkSurfaces). In this network optimal paths must be found to navigate the pedestrian from origin to destination. The model supports the semantic of navigational path similar as in the IFOPT standard. A navigable path is a representation of a detailed pathway that a passenger may take between two surfaces. These start and end surfaces can be the boarding and egress positions in a public transport trip. These boarding and egress positions, also modelled as NodeSurfaces, are the connecting links of this conceptual model the standard data models used in public transport travel advises as described in paragraph 3.3.1. The navigational path describes the pedestrian connection in the public transport advice. In the new conceptual model the pedestrian accessible surfaces (NodeSurfaces) are modelled as objects inherit attributes and behaviour from

nodes in a network. In Figure 26 the classes from network package are purple. In this UML class diagram can be seen that an optimal pedestrian route is modelled as derived from the concept of path and the LinkSurface derived form the concept of a link in a network. Furthermore in Figure 26 the derived relations between NodeSurfaces, LinkSurfaces and the navigational path for the optimal pedestrian route are showed. The necessity to connect public transport with pedestrian routing is illustrated by the possibility to connect green and yellow classes. Connecting boarding and egress locations of public transport vehicles, in a public transport advice, to an optimal pedestrian route is done though the start- en end NodeSurfaces in this optimal pedestrian route. An optimal path is an ordered collection of NodeSurfaces. The start NodeSurface is defined as the first NodeSurface in the sequence of an optimal route and the end NodeSurfaces as the last NodeSurface.

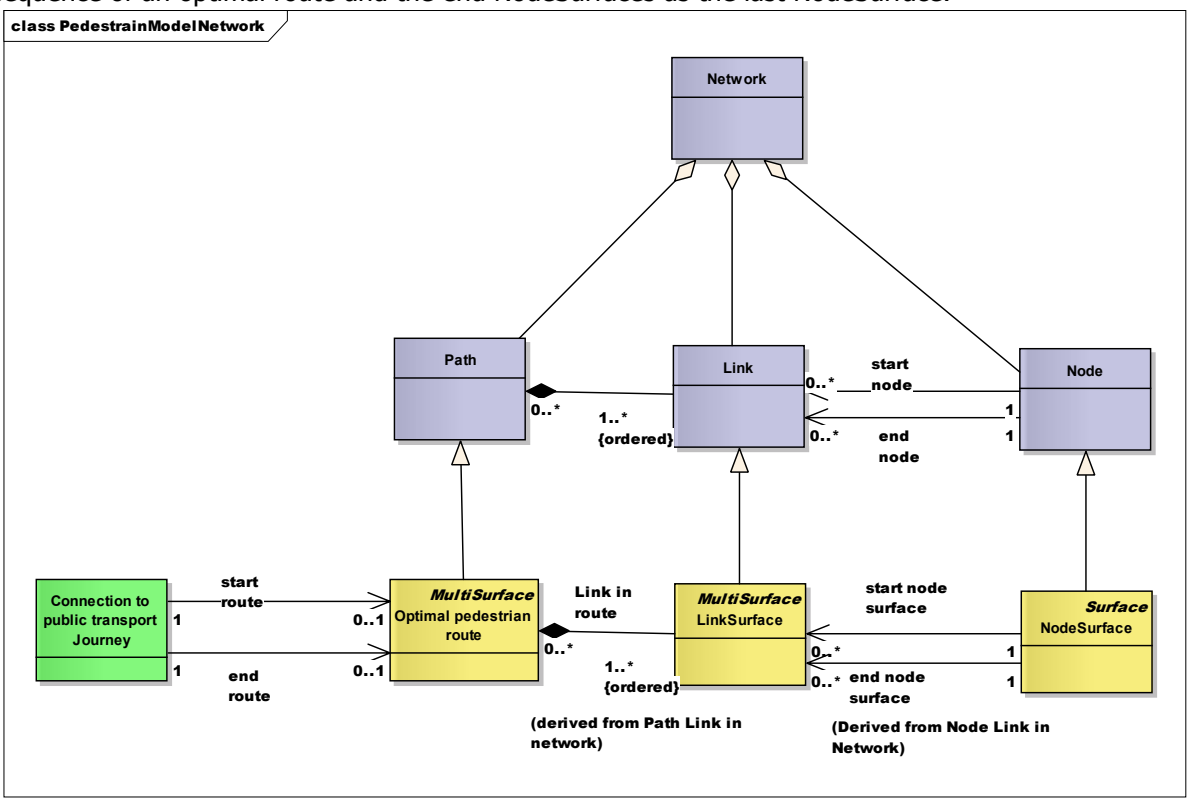


Figure 26 Class diagram surfaces are modelled as objects inherit attributes and behaviour from nodes in a network

During this research an experiment was done with to model 2D routing over surfaces. This experiment was done to get experiences with routing over surfaces using Oracle 11g. This experiment created a basis for route finding in the prototype as described in paragraph 6.2.1. In this experiment a route selection between a University building as origin ("Forum") and destination bus stop ("Droevendaalse steeg, direction Ede") was modelled with surfaces. These surfaces were draw as in Figure 25 and used to construct a network of the surfaces. A layer with touching (Blue) surfaces and a layer with overlapping (Yellow) surfaces where drawn. Then a network was constructed by creating links between in 2D touching or overlapping surfaces with Oracle spatial functions. A navigation path is then an ordered list of links describing the sequence of node surfaces. Figure 27 shows such a navigable path as ordered list over from surface 1 to surface 7 over the surfaces 1, 28 26, 25, 10, 6, 9, 8 and 7 in the pedestrian network. In Figure 27 "Start" is the surface representing the exit of the University building and "End" the boarding position at the bus stop. Near node 28 (step 1) is shown that a sequence in a navigate path is a selection out of all the links from that node. From node 28 the link to 26 is chosen as step 2 in the navigable path. In this example half the perimeter of the surfaces was used to calculate the distance over the surfaces. The surfaces from both drawing layers representing were found in the shortest path. (Overlapping surfaces are having upper case characters in the names in Figure 25 and Figure 27) Please note that in this figure of the network is simplified. All connections are one directorial connections in the model. To simplify the image the drawing software GraphSharp

(see appendix A for software references) combines two one directional arrows between the same nodes in one bi-directional arrow.

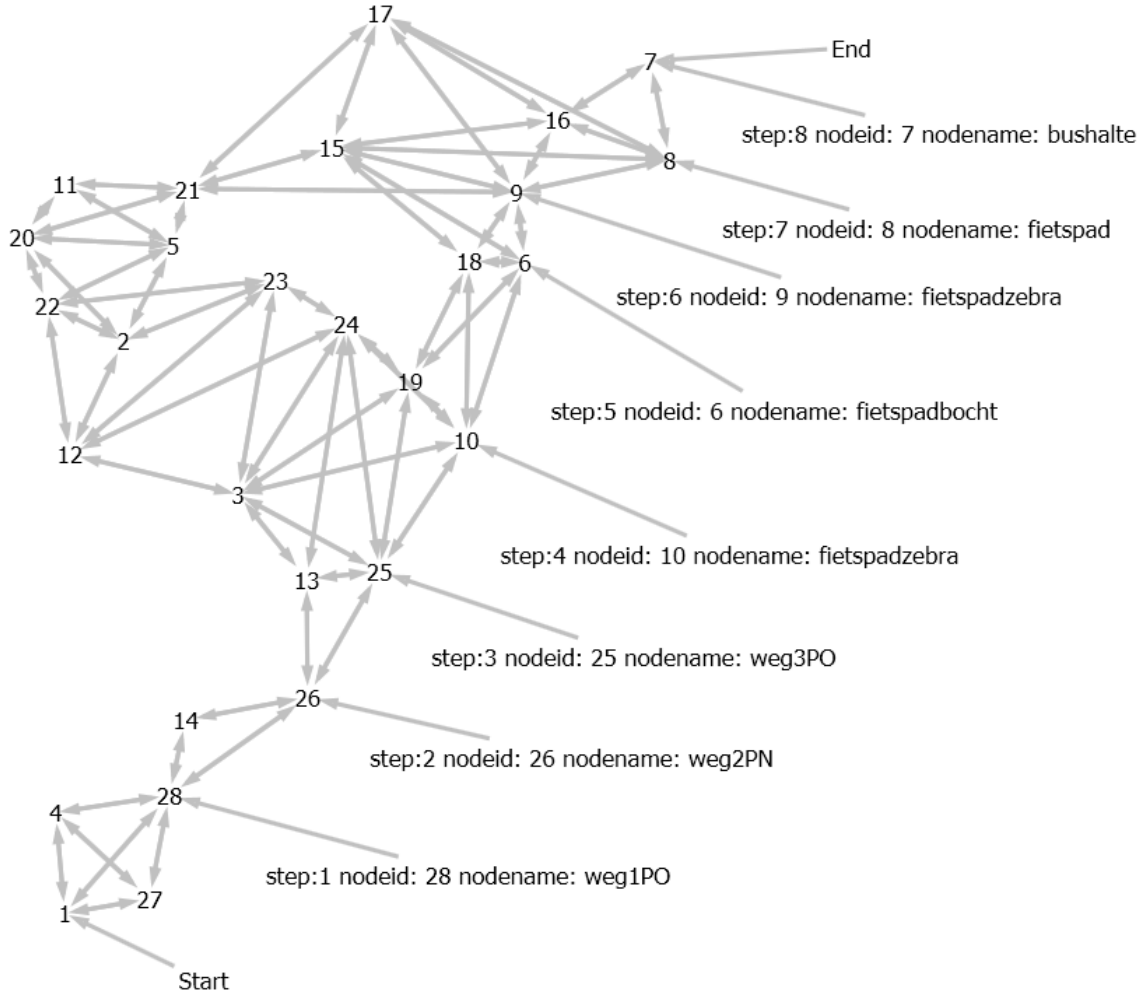


Figure 27 Example of a pedestrian network for 2D routing

5.7 Travellers profile

All objects described in the previous paragraphs of this chapter are based on modelling the scene space, now it is time to focus on the user of this scene space the potential traveller. For this research the individual potential traveller is categorized. This is challenging because traveller route choice behaviour is based on factors as travel times and time pressures, physical efforts, safety and comfort, personal characteristics, and vicinity. These factors are influencing route choices as taking the stairs or an elevator. An example of the differences in individual travellers in seen on escalators some are standing while others walk on escalators. The different behaviours of the individual travellers in this case are base on factors as physical effort, travel time and time pressure, habits, health concerns, and randomness of human behaviour (Lee 2005)

This research takes the individual traveller needs in an account in the pedestrian part of a public transport trip to find an optimal walk route. An optimal route for an individual traveller is based on various preferences. A classification of fast, save, simple or attractive is found for selecting bicycle trips (Hochmair 2004). Other research (Kikiras, Tsetsos et al. 2009) tries to capture even more capabilities of an individual user to support navigational services including cognitive characteristics of users. Detailed modelling these cognitive characteristics is not done in this research, this researched is limited to find an optimal (fastest) pedestrian walking route taking mobility needs and movement speed into

account using a travellers profile. After focusing on modelling of the mobility needs in the next paragraph, the modelling of movement speed is described.

5.7.1 Accessibility in pedestrian route selection

In this conceptual model accessibility is modelled with access types added to LinkSurfaces. The access types of pedestrian spaces in this model are named with names used in the IFOPT document (IFOPT 2007). For this conceptual model no differentiation in made in medical needs, mobility needs, psycho sensory needs or encumbrance needs.

In this model we use a model to optimize walk time and restrict this with individual preferences, it focuses on optimizing travel time. This means that the optimal route in time is calculated. This fastest route differences from shortest route in distance. An example is the usage of a lifts. Lifts are, because of waiting time considered slower for a level change in comparison of a stair but are connecting the levels in a shorter distance. Physical effort is consider is as an individual choice. When an individual considers avoiding particular constructions because of physical effort then in this model it is seen as an accessibility limitation for that user. An assumption is that factors like heavy luggage are influencing this physical effort choice for the individual user. It is considered that that there is a difference in physical effort in going up (stairs, ramps and escalators) or going down and that there are also travellers with problems to take downs stairs/ramps and elevators because of e.g. visibility problems. The relation of individual traveller needs and the possibility to take a link of a particular type is shown in Table 1. In this table is defined that a perfect flat surface with the access type "No restriction" is accessible for all kind of travellers and a lift is not accessible for travellers "Averse to lifts". Furthermore vertical direction is important except for single step (e.g. a single tread to use a sidewalk) and lifts.

	No restriction	**Single Step (Up or Down)**	**Ramp Up**	**Ramp Down**	**Stair Up**	**Stairs Down**	**Lift**	**Escalator Up**	**Escalator Down**
Wheelchair	Y	N	Y	Y	N	N	Y	N	N
Step Free	Y	N	Y	Y	N	N	Y	Y	Y
Averse to lifts	Y	Y	Y	Y	Y	Y	N	Y	Y
Escalator free	Y	Y	Y	Y	Y	Y	Y	N	N
Luggage heavy	Y	N	N	N	N	N	Y	Y	Y
Luggage medium (no stairs ramps up)	Y	Y	N	Y	N	Y	Y	Y	Y
No down stairs/ramps /escalators	Y	N	Y	N	Y	N	Y	Y	N

Table 1 Individual needs related to Link access types.

The relation of individual traveller needs and the possibility to take a link of a particular type is directly used in the conceptual model where a traveller has one or more mobility needs which are related to the possibility to use a LinkSurface. The accessibility type is related to LinkSurface because accessibility is depending on the type of NodeSurfaces and the vertical direction the traveller has to overcome. Figure 28 shows these relations in a UML Class diagram.

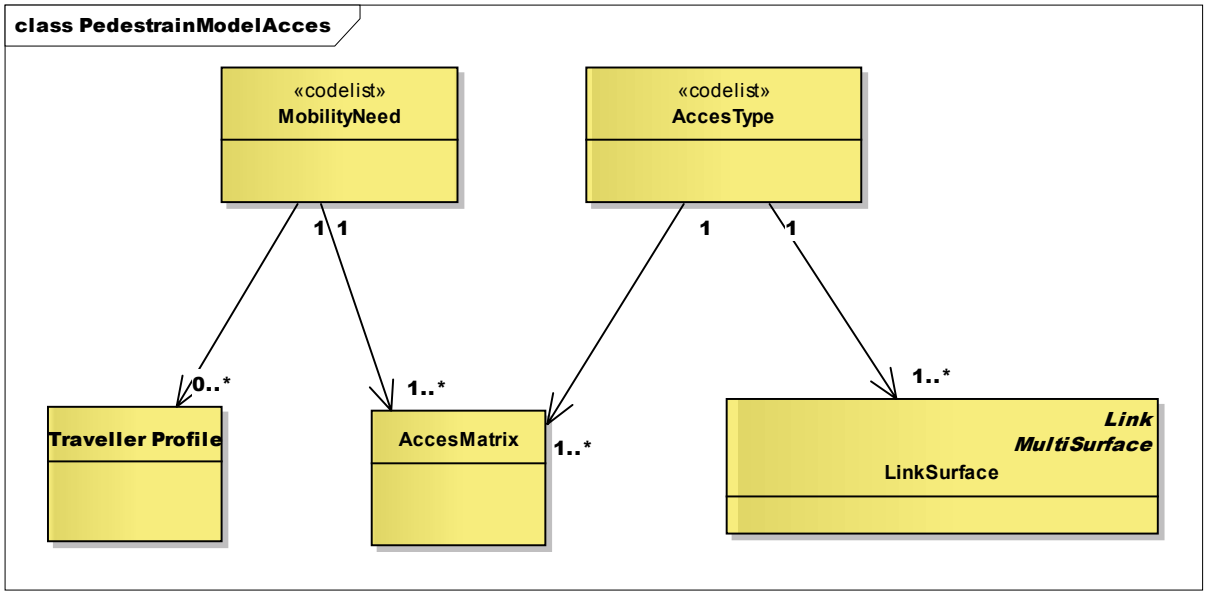


Figure 28 Class diagram relating traveller's profile to LinkSurfaces for individual needs

5.7.2 Movement speed in pedestrian route selection

The model supports the concept of adding walk durations to links. The model is designed that walk durations in can be added to LinkSurfaces in the model. When measured walk duration values are know for objects modelled with LinkSurfaces in a scene spaces, these values could be added to the LinkSurfaces. In this research no measurement of walk duration is done to determine the walk durations added to LinkSurfaces. A model is made to estimate the walk duration for a link based on walk speeds found in literature (Lee 2005). Table 2 summarizes some free speeds measurements on different types of infrastructure which include walkways or passageways, stairways and escalators. Horizontal speeds of one-directional flows are given in Table 2. The data shows that speeds on walkways or passageways are higher than those involving vertical movement on stairways and escalators. In additions, speeds in the descending directions are higher that those in the ascending one. The differences in values are based on the fact that lower speeds are observed on stairways with larger slopes and higher risers. The values from literature are measured in different parts of the world; it can be assumed that these speeds are also based on cultural differences.

From literature actual walk speeds are defined in the model. In this model a simplification is made and it is assumed that the horizontal distance in combination with the type of node surface and the vertical direction is a measure for the actual walk duration of a surface. The steepness of a ramp is not taken into consideration. The walk duration is considered a constant although every traveller knows that it takes longer to use stairs at an interchange during rush hours.

	Bus terminal USA 1970	Metro station UK 1991	Metro station HongKong 2000	Train station Nederlands 2005	Model values[6]	
Walkway or Passageway	1.35	1.47	1.37		1.4	+/- 50%
Stairway up	0.51-0.57	0.59	0.86	0.68	0.7	50%
Stairway down	0.67-0.77	0.67	0.97	0.77	0.8	+/- 50%
Escalator up	-	0.84	0.89	0.82	0.8	None[7]
Escalator down	-	1.00	1.05	0.88	0.9	None[7]
Ramp down					1.6	+/- 50%
Ramp up					1.2	+/- 50%
Lift up					60[8]	None
Lift down					60[8]	None
Steps					1.4	+/- 50%

Table 2 Walk speed from literature (Lee 2005) **and model values (in m/s horizontal speed)**

Furthermore in the model 3 types of travellers are considered:

1) Slow
2) Average
3) Fast

In "Figure 29 Class diagram accessibility and speed are modelled similar" can be seen that the amount of speed types is flexible and these speed types are similar related to travellers profile and a LinkSurfaces as the mobility needs. In the model the assumption[9] is made that fast travellers can increase there walk speed with 50 percent on walkways, stairs and ramps and slow travellers are using a speed with 50 percent less of the average traveller on these objects. The values for these speed types relegated to the LinkSurface types are showed in Table 3. As stated before this estimation of the walk times and profiles is only done to validate the capabilities of this new model. Working with observed values for LinkSurfaces would take cultural different and local circumstances into account.

To support various types of walk duration methods for different kind of NodeSurfaces, the concept of calculation method numbering is introduced. Method 1 is using half the perimeter of a NodeSurface in combination with the speed for a specific kind of NodeSurfaces and traveller profile combination. As described in paragraph 5.5 lifts are modelled in a specific way. Using this method 1 half the perimeter of "back wall" of a lift would be a measure for the duration of a lift usage. A method using the average waiting time and the number of floors and in between stops would be more appropriate. Modelling waiting and usage time of lifts is considered outside the scope of this model. In this model is a specific method created for lifts (method 2). Method 2 could implement a complex function returning durations of lift usage. In this model is lift usage time is set to 60 seconds for all up or down lift usage.

[6] in m/s horizontal speed, except for lift including speed profile Slow/Fast correction for speed profiles used in this model
[7] walking on escalators is not considered
[8] In seconds (fixed)
[9] Research into these walk speed and lift profile values is considered outside the scope of this research.
[11] in m/s horizontal speed, except for calculation method 2 (lift) is fixed seconds

Accessibility type	Vertical direction (for pedestrian)	Normal[11]	Slow[11]	Fast[11]	Method of calculating duration for the NodeSurface
Escalator Down	Down	0.9	0.9	0.9	1
Escalator Up	Up	0.8	0.8	0.8	1
Lift	Level	60	60	60	2
No Restriction	Level	1.4	0.7	2.1	1
Ramp Down	Down	1.2	0.6	1.8	1
Ramp Up	Up	1.6	0.8	2.4	1
Stairs Down	Down	0.8	0.4	1.2	1
Stairs Up	Up	0.7	0.4	1.1	1
Steps	Level	1.4	0.7	2.1	1

Table 3 Speed profile values mapped to accessibility types (m/s horizontal speed except lift values in seconds)

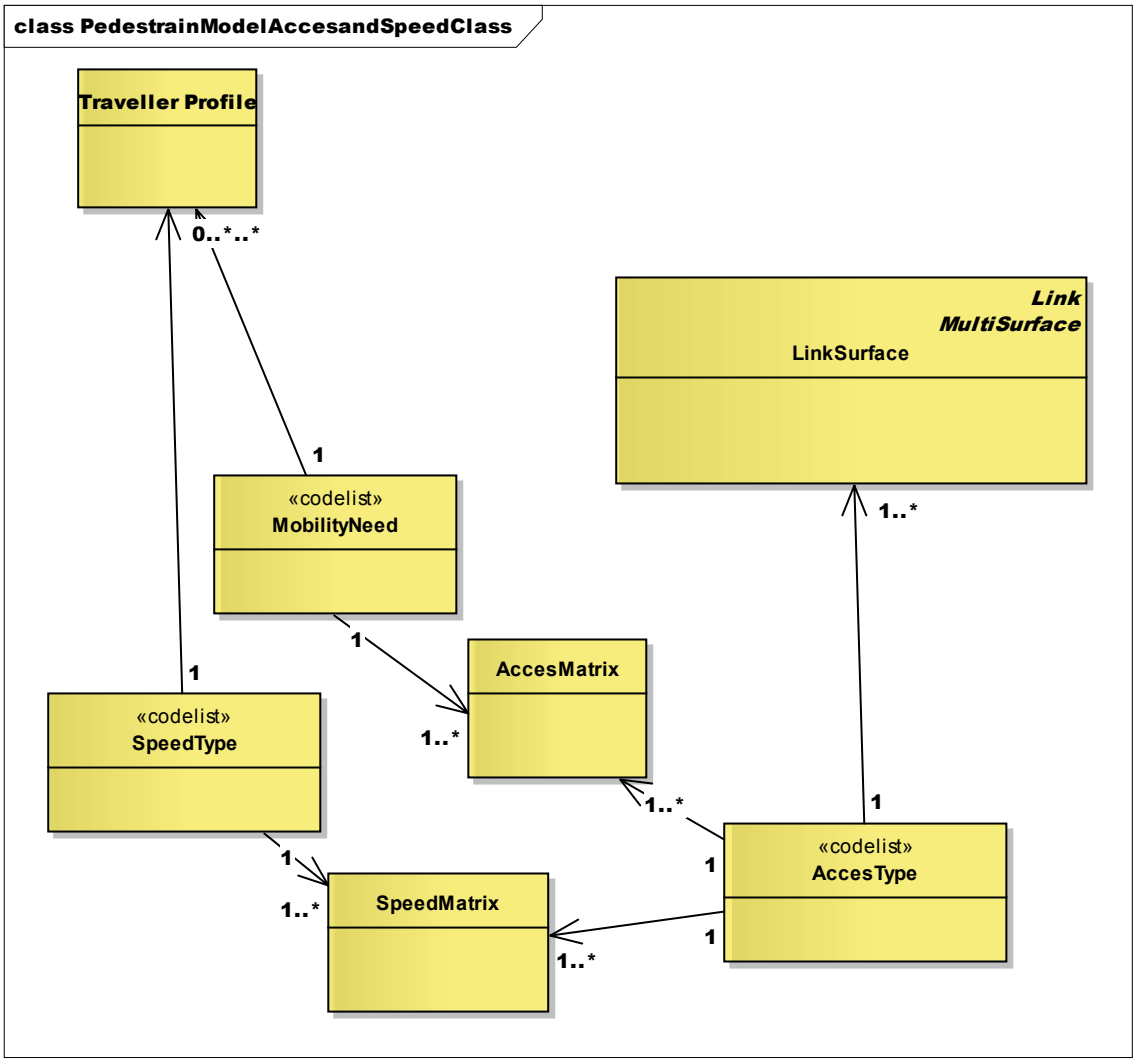


Figure 29 Class diagram accessibility and speed are modelled similar

5.8 Optimal pedestrian route calculation

The classes for optimal path calculation where showed in Figure 26. Using these classes in combination with the classes related to the travellers profile, optimal pedestrian route calculation can be done to

enhance a standard public transport travel advice with pedestrian routing. To enhance a standard public transport travel advice with walk routing for a traveller having a specific travel profile the following steps must be done:

1) Optimal navigational path calculation starts with deriving the start en origin from the walk parts from a public transport travel advice. These start en origin points must be matched on NodeSurfaces describing these points. (E.g. IFOPT Boarding positions, entrance or public places). Connecting boarding and egress locations of public transport vehicles is done though the start- en end NodeSurfaces in an optimal pedestrian route.

2) The traveller profile is composed. The walk speed type in combination with the individual needs from the traveller is defined for this traveller.

3) With the individual needs from the traveller profile, a sub selection of all LinkSurfaces is selected containing all LinkSurfaces this traveller can possibly use in an optimal path.

4) For the set of LinkSurfaces selected for the traveller profile walk durations are assigned to each LinkSurface according to the speed type in this profile. This duration for each LinkSurface is calculated as described in

5) After construction of this set of LinkSurfaces, fitting this traveller profile the fastest path (shortest path in time) is must be calculated between start and origin NodeSurface as navigational path for this walk part.

6) Different navigational paths with the same total walk duration can be found as fastest path. In this case both are seen as optimal solution and one of these navigational paths is selected as optimal pedestrian route. It is also possible that for this individual user the destination is not reachable.

7) The steps 1 till 6 are repeated for each walk part of a public transport travel advice. If a destination is not reachable a new public transport advice must be generated avoiding this walk part. If because of pedestrian constraints the destination of the walk part is not reachable in time to connect to public transport also a new public transport advice must be generated using the new departure time at this boarding position. (and step 1 till 6 repeated)

8) After optimal navigational path are found for all walk parts of a public transport advice, a public transport advice is created including pedestrian routing. The pedestrian routes found for this travel advice can be used for textual walk times and accessibility and route information.

6 A PROTOTYPE FOR PEDESTRAIN ROUTING

For the prototype an object-relational model is created for representing geometries in Oracle (Version: Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production). This model stores an entire geometry for a surface in the Oracle native spatial data type for vector data, SDO_GEOMETRY. This object-relational model corresponds to a "SQL with Geometry Types" implementation of spatial feature tables in the OGC ODBC/SQL specification for spatial features with is in line with the Simple feature specification for SQL which was described in paragraph 3.4.1. In line with the conceptual model for storage of the surfaces of a node surface a SDO_GEOMETRY of type surface is used (SDO_GTYPE=3003). Next to the Oracle database the following tools where used to create this prototype:

1) Google SketchUp
2) Enterprise architect
3) FME
4) SQL developer
5) Google earth
6) C# visual studio + GraphSharp l

In appendix A references are made to sites describing these tools. To run these tools adequate hardware was available. A desktop pc (HP Compact dc5700 Microtower with dual core Pentium 3.00 GHz processors and 600 GB hard disk) was used for this research. Licences for the used software to run on this desktop pc where supplied by GMDC. With an internet connection an Oracle 11 g server at GMDC was used for implementing the prototype.

In Figure 30 the activities for the construction of the prototype are visualized in a (data) flow diagram. In this figure the processes are shown in the ellipses, the artefacts as blocks. A special symbol is used for the Oracle database. In this dataflow diagram a separation of processes and artefacts can be seen for getting the pedestrian data model in the Oracle database and operations on this data model to get optimal paths out of the Oracle database. This separation is used to describe the construction and usage of the prototype in this chapter. After a describing how the data is modelled to store the pedestrian data in the database (section 6.1), a description is given how the optimal routes are derived and visualized with this data (section6.2). The table definitions used for storing and visualisation of this pedestrian data are found in appendix K till N.

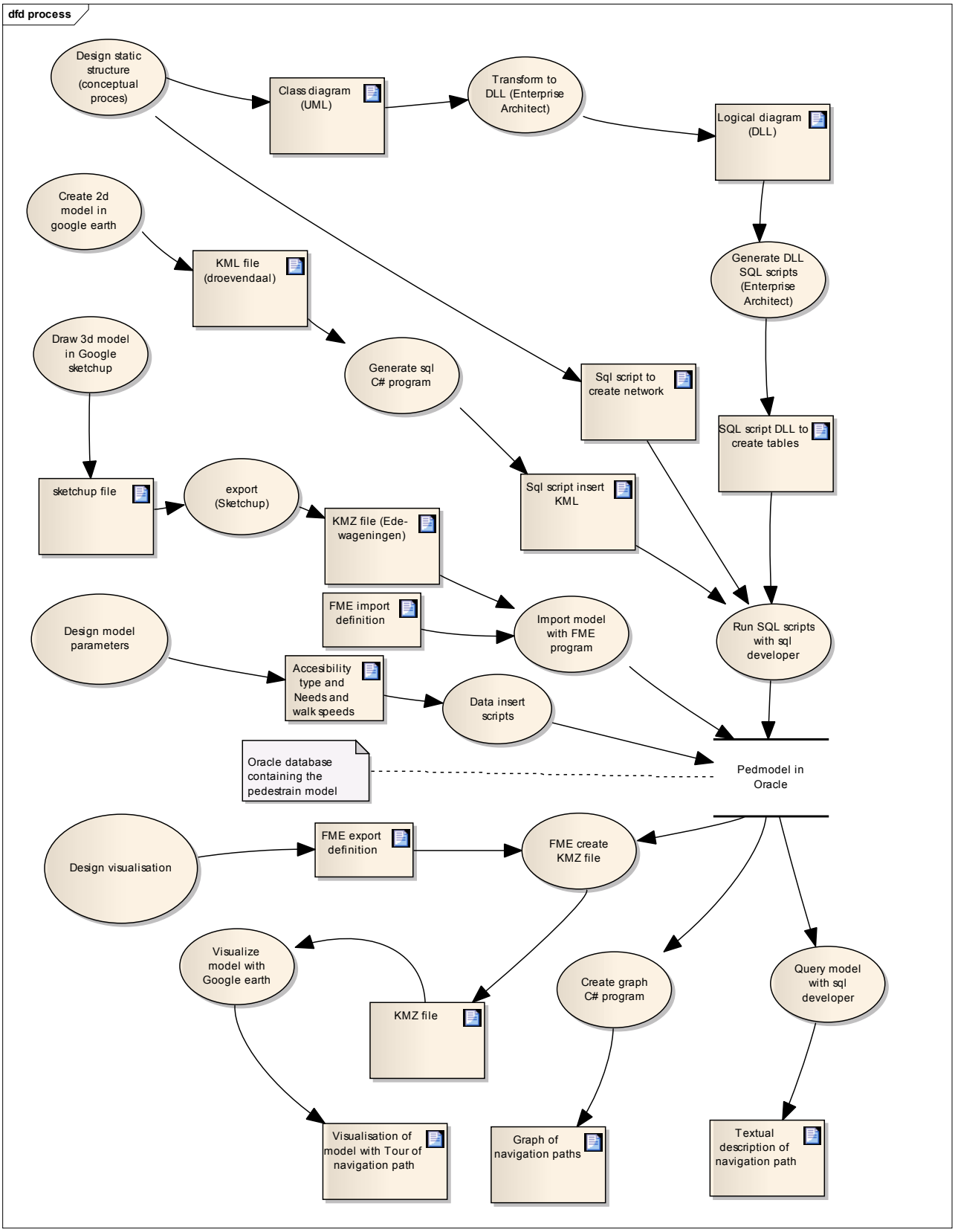


Figure 30 Flow describing the construction of the prototype

6.1 Storing the pedestrian prototype data in the Oracle database.

As described in the previous chapter network and surfaces are the basis for the conceptual model. The LinkSurfaces are combining attributes and behaviour from both and are related to the potential accessibility and walk speed of specific travellers. In this section is described in detail how storage in Oracle 11g is done following the conceptual model as described in the previous chapter.

6.1.1 Modelling the network and surfaces

After the experiment described in paragraph 5.6 to test the shortest path calculation capabilities for Oracle spatial, construction of the prototype started. First a simple model to validate the use case "Locating the entrance (s) of a public transport place in relation to the surroundings" was created. For this model first location origin "Forum" and destination bus stop "droevendaalse steeg (direction Ede) was used. In Google earth the NodeSurface where drawn between the origin and destination. These NodeSurfaces where exported as KML files. The model used for the experiment described in paragraph 5.6 was improved and model bus stops were modelled in detail using Google SketchUp and placing the SketchUp models on Google earth (Figure 31). For the other use cases a more complex 3D model the second location train station "Ede Wageningen" was used and a model in Google SketchUp was created with a ramp, stairs lifts, an underpass (tunnel) and boarding positions. This model was exported to a kmz file. To do validation of the use cases the bus stop "Ede Zuid" was selected as origin and the first boarding position near the stairs at "platform 3 direction Utrecht", as destination.
The prototype uses a geodetic system (WGS84) as reference system and altitude relative to the ground for the Z direction. This is done because no transformation is needed when capturing data from Google Earth and projecting data back in Google Earth in the prototype where altitude mode relative to the ground in the KML files is used. For distance calculation in 3D space a local system with similar units of measures (e.g. RD "Rijksdriehoek") in all 3 directions is more suitable. 3D Distance calculation with such a system is supported by Oracle (Murray 2010). Google SketchUp and the generated DAE models used KML files are also containing a Cartesian coordinate system with its origin as the anchor point to place the model on the earth surface. Furthermore the model is vector based and uses as basis the elements from the Simple Feature Implementation Specification for SQL (OGC 1999) as described in paragraph3.4.1.

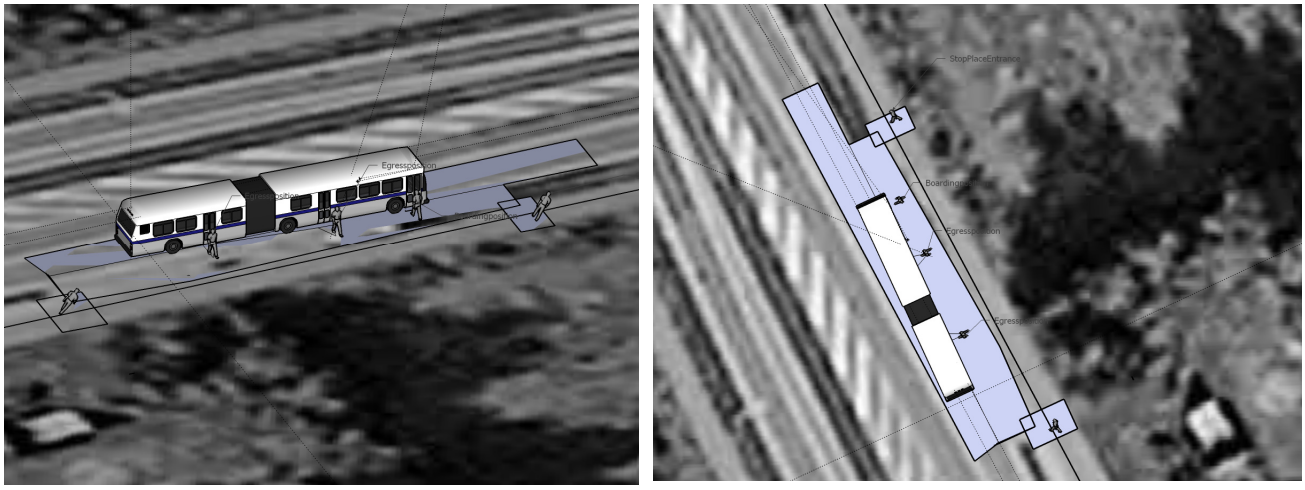


Figure 31 Detailed stop modelling by placing SketchUp models on Google earth (Bus stop Droevendaal one sides of the road direction Ede)

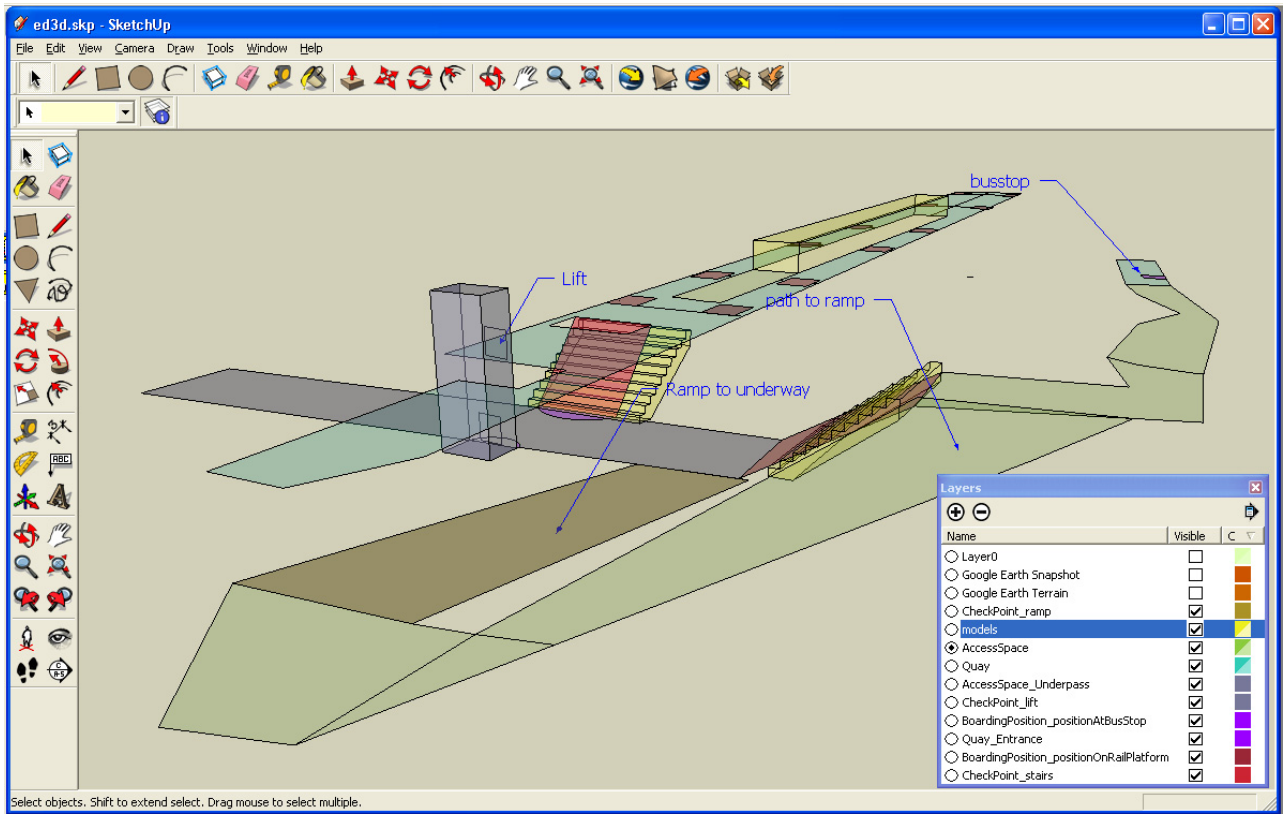


Figure 32 NodeSurfaces for studied location Ede-Wageningen in Google SketchUp

In Oracle standard functions to create networks and network analysis exists, it has network modelling capabilities, include schema objects and an application programming interface (API). The schema objects include metadata and network tables. The API includes a server-side PL/SQL API (the SDO_NET and SDO_NET_MEM packages) for creating, managing, editing, and analyzing networks in the database, and a middle-tier (or client-side) Java API for network editing and analysis. (Murray, An et al. 2007). In Oracle 11g an approach that divides large networks into manageable partitions exists. This approach loads only network parts when needed, keeping memory usage on the computer that does network calculations lower than in previous Oracle versions were the whole network was loaded in memory. This functionality called "load on demand" is especial useful for large networks (more than 5000 nodes). The prototype network is small therefore this new "load on demand" functionality is not used. A network model consisting of node link and path tables exists. In the Oracle 11g documentation no UML diagram is given of the relation between the various network objects. Figure 33 shows the Java Classes and Interfaces for Network Data Model as described in the documentation of previous version of Oracle (Murray, An et al. 2006). The figure gives a good overview of the classes available in the Network package and the relations. Please note that in this figure the relation between Node and Link is modelled that 2 nodes are always belonging 1 Link (Node: Link as 2:1). This is wrong because one node can be used in more links, therefore a Node: Link relation is at least 2:1..n. Furthermore this notation does not differentiate the start and end role of a node in a link. In the pedestrian model these difference are essential because they are used to model the difference between going up or down. Therefore these roles are explicit modelled in the conceptual model Figure 26. Same applies for the relation Path – Link relation (Path: Link as 1:n in Figure 33). A Link can exits in more than one path (Path: Link as n:n). In the prototype a "path_link_table", named "RouteElements" exists used as connecting table to model this n:n relation. (Route: RouteElements: LinkSurface as N:1:N) This table is also contains the ordering (sequence) of LinkSurfaces in an optimal Route. This ordering of links is modelled in the pedestrian model (Figure 26) but not in Figure 33.

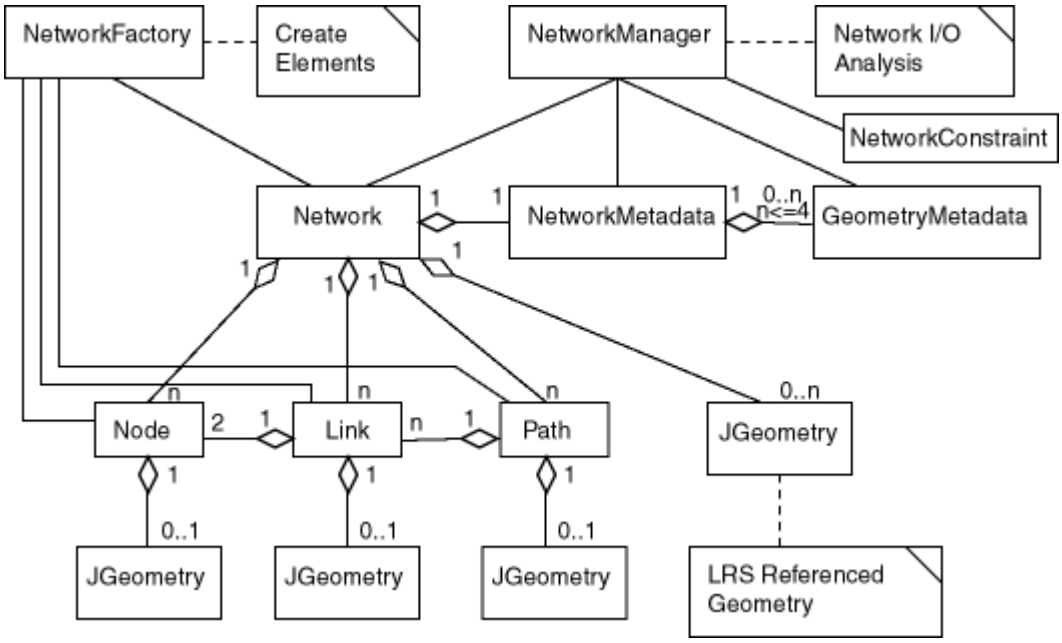


Figure 33 Java Classes and Interfaces for oracle Network Data Model (Murray, An et al. 2006)

Standard SQL and Oracle specific PL/SQL are used in combination with Oracle supplied PL/SQL interface functions to create and analyse the pedestrian network model.

Parameter	Description	Usage in pedestrian model
Network	Network name.	PEDMODEL
no_of_hierarchy_levels	Number of hierarchy levels for links in the network	1 (no levels are used)
is_directed	A Boolean value. `TRUE` indicates that the links are directed; `FALSE` indicates that the links are undirected	True (direction is important in the model)
node_with_cost	Cost to be included in the node	Cost in NodeSurfaces are the walk around or hidden path walk duration for surfaces where direction is not important.
node_table_name	Name of the node table to be created	NodeSurface
node_geom_column	Name of the geometry column in the node table.	networkGEOMETRY Field Not used (could be used for visualisation purposes, value is always NULL).
node_cost_column	Name of the cost column in the node table	Field called Delay is not used, value is always 0 (could be used to model extra time penalty for walk duration over a surface, not related to direction).
link_table_name	Name of the link table to be created.	LinkSurface
link_geom_column	Name of the geometry column in the link table.	networkGEOMETRY (Not used could be used for visualisation)
link_cost_column	Name of the cost column in the link table.	Duration Cost are the walk around or hidden path walk duration for start_NodeSurface and end_NodeSurface
path_table_name	Name of the path table	Route
path_geom_column	Name of the geometry column in the path table	networkGEOMETRY (not used could be used for visualisation)
path_link_table_name	Name of the path-link table to be created	RouteElements

Table 4 parameters for create_sdo_network

In PL/SQL the function SDO_NET.CREATE_SDO_NETWORK was used to create the tables NodeSurface and LinkSurface and tables (route and routeElement) to store the result navigational path table. Table 4 is describing used parameters in the function call:

```
SDO_NET.CREATE_SDO_NETWORK('PEDMODEL',1,TRUE,'NodeSurface','networkGEOMETRY'
,
'distance','LinkSurface','networkGEOMETRY','delay','route','networkGEOMETRY'
,'routeElement');
```

The KML files with the NodeSurfaces where parsed with a C# program and converted to SQL statements to insert the data in the table NodeSurface. To convert the data from KML to Oracle surfaces the Oracle utility function "FROM_KMLGEOMETRY" is used. The more complex kmz file created with Google SketchUp is imported in Oracle with a FME job. In the kmz file generated with SketchUp polyhedrons are modelled as multiGeometry. For the model NodeSurfaces must be created consisting of Oracle surfaces (SDO_GTYPE=3003). A conversion is FME (Deaggregate) is done to create surfaces out the multiGeometry objects in the kmz files. The Google SketchUp layer name is used in combination with the sequence number of the surface in the layer (part number). The definitions of the table, named polygons, use to temporary store these surfaces is given in appendix N. The network was created with the Oracle function to create a network. It was found out that an SDO_NETWORK only works with geometric objects containing points and lines. The surface (SDO_GTYPE=3003) is not supported by the oracle SDO_NETWORK. To overcome this limitation an extra field GEOMETRY3D was added to the NodeSurface table to contain the surfaces for the pedestrian model. In the SDO_NETWORK the lines and points geometry columns "networkGEOMETRY" are not used it could later used only for visualisation if presentation with 2D lines instead of surfaces is needed when the model is generalised in presentation. Furthermore the vertical direction (horizontal, up and down) is important in the pedestrian model. To derive this vertical direction in the links the average Z value of the NodeSurface is extracted from the surface and stored in a extra field Z. This works because NodeSurfaces are not curved. The following SQL statement set the Z value for the NodeSurfaces:

```
Update NODESURFACE c
set Z= (select round ( (min(t.Z) + max (t.Z))/2 , 0)
  FROM NODESURFACE  ,
   TABLE(SDO_UTIL.GETVERTICES(c.GEOMETRY3D)) t
  where node_id = c.node_id ) ;
```

[13] This is a limitation of automatic deriving connections between node surfaces with the standard (limited 3D intersection) capabilities of Oracle 11g. Manual inspection and correction, improvement of this algorithm, or cleaning and modifying source data is needed to avoid this shortcoming.

The LinkSurfaces are derived from the NodeSurfaces. All NodeSurfaces are related to each other (cross join) and a selection is made if there is any spatial connection (touch, overlap, intersect) in 3D, between the surfaces. (This models the relation described in 5.3: "Start_NodeSurface end end_NodeSurface must be connected. The model restricts this multiSurface to a surfaces consisting only two surfaces with only one connecting border. No disconnected surfaces are allowed in this multiSurface. If surfaces are related to each other (even a touch with only the corners in 3D[13]) then a connection between is assumed and a link surface is created for these 2 surfaces.
This is done in the following way.

```
INSERT
INTO   linksurface
       (
              link_id,link_name,start_node_id,end_node_id,active,Zdirection
       )
SELECT rownum,a.node_name || ' -> ' || b.node_name || ' ' ||
       CASE
              WHEN a.Z < b.Z
              THEN '(Up)'
              WHEN a.Z > b.Z
              THEN '(Down)'
              ELSE ''
       END
       CASE
      ,a.node_id,b.node_id,'Y',
                     CASE
                            WHEN a.Z < b.Z
                            THEN 'U' -- up
                            WHEN a.Z > b.Z
                            THEN 'D' -- down
                            ELSE 'L' -  horizontal (level)
                     END
                     CASE
                            FROM   NODESURFACE a,NODESURFACE b
                            WHERE                           SDO_GEOM.RELATE(
a.GEOMETRY3D,'ANYINTERACT',b.GEOMETRY3D,0.000000000000005 )= 'TRUE'
                            AND    a.node_id
```

6.1.2 Modelling accessibility information

In the classes modelled for accessibility in (showed in Figure 28) are used as basis for the accessibility tables. From the class model a transformation is done to create a data model with enterprise architect. To illustrate this model transformation, the class model showed in Figure 35 is transformed and the result are the table definitions (DDL) showed in Figure 36. In appendix L are the resulting scripts for these tables given.

In this data model 3 tables are define

1) Accestype only containing a unique identifier and the description.
2) Mobility Need only containing a unique identifier and a description of the mobility need
3) AccesMatrix a relation which mobilityneed can make use of which accestype. It is a direct representation of Table 1 on page 45. It contains only entries for the "Y" cells in this table.

By pivoting the contents of the access matrix table and using the other two tables the content of Table 1 on page 45 can be derived. This is showed in Figure 34.

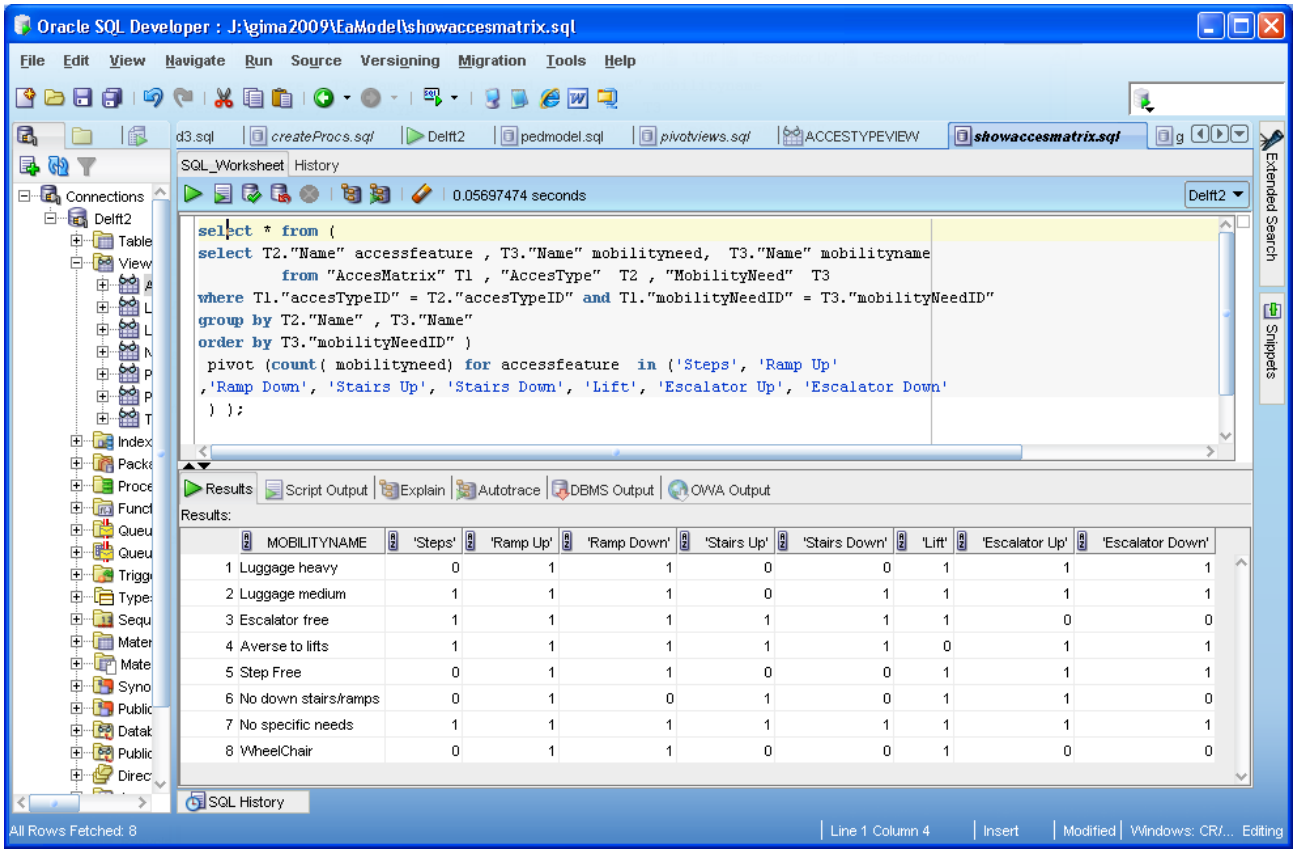


Figure 34 Pivot SQL to show accessibility matrix

In the LinkSurface table a field to store a reference to the accestype of the link is added. NodeSurfaces are not directly connected to accestype (and therefore not showed in Figure 35); the vertical pedestrian direction is modelled with LinkSurfaces (having a start_NodeSurface and an end_NodeSurface).

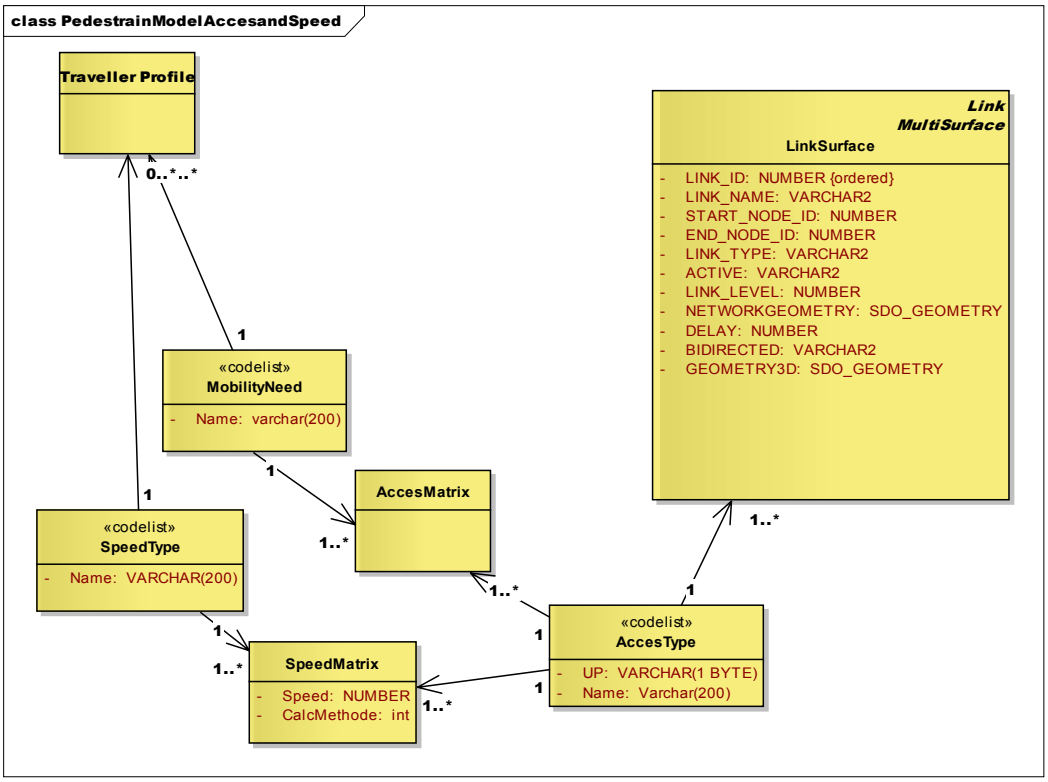


Figure 35 Class model relating accessibility and speed aspects from traveller profile to LinkSurface.

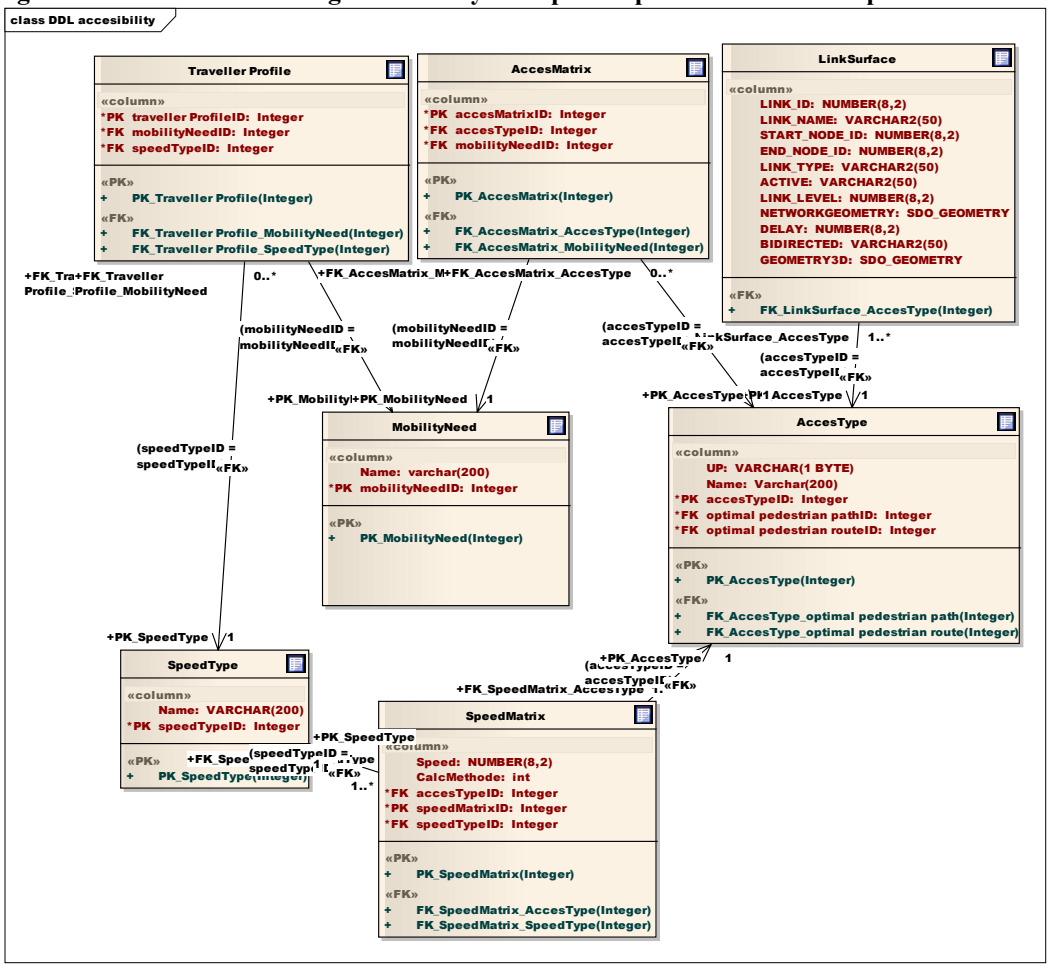


Figure 36 DDL to create tables relating accessibility and speed aspects from traveller profile to LinkSurface.

6.1.3 Modelling walk duration

Observed walk duration for all LinkSurfaces for all travellers profiles could be stored in the model. In the prototype standard speed are stored in the walk speed matrix (see Table 2). This speed matrix table is used to calculate the duration per LinkSurface. The walk distance for a LinkSurface is estimated by taking halve the value of the perimeter of the start_NodeSurface not taking the height into account. Distances are measured in 2D for 3D surfaces for geodetic data in Oracle. If the data is geodetic (geographic 3D), the distance computations are done on the geodetic surface (Murray 2010). This distance is used to calculate the duration based on the relevant value in het speedmatrix. The speed matrix works similar as accessibility matrix. Standard walking speeds values are defined for every accestype and for every speed profile (fast, normal, slow). Duration for the start_NodeSurface of a LinkSurface is calculated with the methods specified in paragraph 5.7.2:

Duration = Distance (horizontal meters) * horizontal speed
For lifts (fixed duration= #units (e.g. levels) *duration (lift)

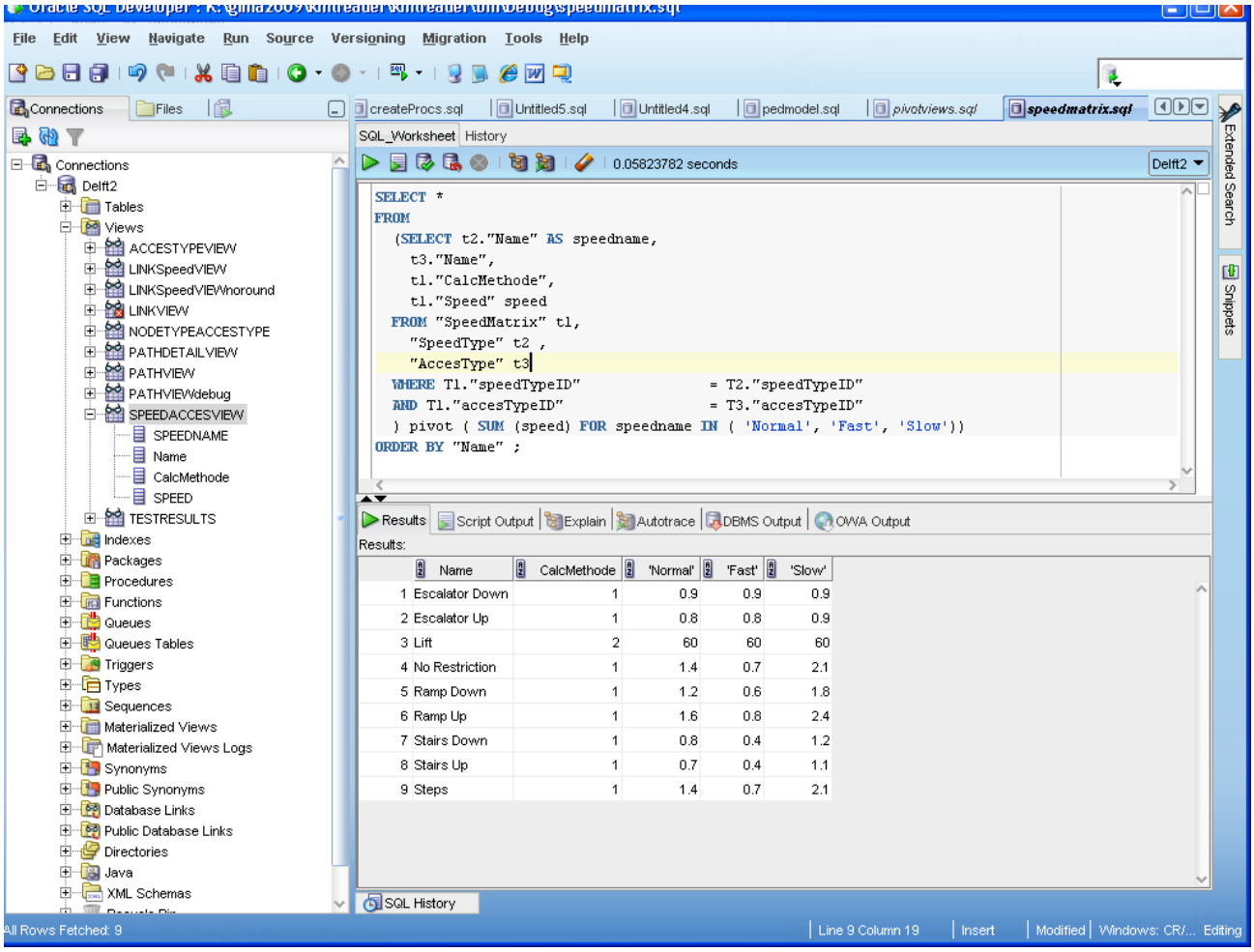


Figure 37 Pivot SQL to show speed matrix

The model supports to persist duration for individual LinkSurfaces. But for this research the walk duration of a NodeSurfaces is calculated from the distance. The distance from the link over the

NodeSurfaces is estimated by taking half the perimeter[14] of the start_NodeSurface. For lifts a fixed duration of 60 seconds is taken[15].

A view (linkspeedview) to get duration per link for every speedtype exists containing the following SQL logic to determine the duration for the start_NodeSurface (T2):

```
SELECT t1.link_id,
         --various attribute fields not relevant to speed.
         CASE t4."CalcMethode"
         WHEN 1 THEN  ((SDO_GEOM.SDO_LENGTH(t2.GEOMETRY3D,0.5, 'unit=M'))/2)
* t4."Speed"
        WHEN 2 THEN t4."Speed"
         ELSE 0
        END,
        t2.GEOMETRY3D


  FROM   linksurface t1,
         nodesurface t2,
         nodesurface t3,
         "SpeedMatrix" t4
  WHERE  t1.start_node_id = t2.node_id
         AND t1.end_node_id = t3.node_id
         AND t4."accesTypeID" = t1.link_level;
```

[14] This simple estimate has limitations, the shape of the NodeSurface influences this estimate, e.g half the perimeter of a star shape differs from half the perimeter a box shape used as estimate for the walking distance over a NodeSurface..

[15] In this research taking the lift from the under passage to the platform or visa versa is set to 60 seconds. This value is taken as example no steps are taken in this research to measure or find appropriate value from literature for the lift duration.

6.2 Operations on the pedestrian model.

The main purpose of the conceptual model is optimal pedestrian route calculation for the traveller following the method as described in paragraph 5.8. In the prototype these optimal paths are called pedestrian routes. When these optimal paths are found they are temporary stored in specific tables. To analyze these optimal routes visualisation of these routes was done with Google Earth. This sections first explains in detail the approach and relevant the Oracle PL/SQL statements to find and store the optimal paths. After this explanation is described how these optimal paths are converted to kmz files to visualize in Google Earth.

6.2.1 Finding optimal routes.

A stored procedure MAKEROUTE is developed during this research to calculate the routes according to the travellers needs and speed profile. This procedure performs the steps 3, 4, 5 and 6 as described in the conceptual model in paragraph 5.8. A full listing of this PL/SQL MAKEROUTE procedure is found in appendix I. The procedure accepts four parameters, the origin and destination of the route to calculate, the id of the mobility need of the traveller and the speedtype (id) of the traveller.
In the prototype route calculation is done in 3 steps with the MAKEROUTE procedure:

1) The links available for the specific traveller profile and speeds are set to active
2) The network is loaded in memory
3) A shortest route is calculated.

1. Set links active and speed for current traveller.

The links available for the traveller are set to active with the following PL/SQL statements in the MAKEROUTE procedure:

```
Update linksurface
set active = 'N',
delay =0 ;

Update linksurface t1
set active = 'Y'
,
delay  = (select "Timeduration"   from "LINKSpeedVIEW" where "speedTypeID" =
speedtype and "LINK_ID"= t1.link_id)
where link_level in (select "accesTypeID" from "AccesMatrix" where
"mobilityNeedID" =needtype) ;
```

The concept of having more need types e.g. "averse to lifts" and "heavy luggage" for one traveller, is not supported in this procedure, although this would be easy to implement by changing the parameter of the MAKEROUTE procedure to an array of needs and to test against this array in the where clause of the "update linksurface" statement.

2. Load network in memory

```
SDO_NET_MEM.NETWORK_MANAGER.READ_NETWORK('PEDMODEL','TRUE');
```

A network memory object in virtual memory cache containing all network objects of the "PEDMODEL" is loaded in memory. This "PEDMODEL" network was created with the procedure create_sdo_network described in paragraph 6.1.1. (Table 4 on page 54 gives the configuration of this network). The name of the network is the first parameter of the SQL statement to load command:

```
SDO_NET_MEM.NETWORK_MANAGER.READ_NETWORK('PEDMODEL','TRUE');
```

The second parameter specifies that the network memory object is updatable. TRUE is used to specify that the network memory object is writable and editing operations in the cache can be written back to the database. Although for the performance of network functions it is better to use a read-only network (Murray, An et al. 2007), in the prototype the write back functionality is used to store the resulting optimal routes.

3. Calculate shortest route

In Oracle the shortest path can be calculated with supplied shortest path functions. Four different shortest path methods are available in the SDO_NET_MEM package. These functions are summarized in Table 5.

SDO_NET_MEM.NETWORK_MANAGER.SHORTEST_PATH	Returns the path ID number of the shortest path (based on the A* search algorithm, and considering costs) between a start node and a goal (end) node.
SDO_NET_MEM.NETWORK_MANAGER.SHORTEST_PATH_DIJKSTRA	Returns the path ID number of the shortest path (based on the Dijkstra search algorithm, and considering costs) between a start node and a goal (end) node.'
SDO_NET_MEM.NETWORK_MANAGER.TSP_PATH	Returns the path ID number of the most efficient (in cost or distance) path that includes all specified nodes, that is, the path that solves the "travelling salesman problem" (TSP) for the specified set of nodes.
SDO_NET_MEM.NETWORK_MANAGER.ALL_PATHS	Returns all paths between two nodes, limited by cost, depth and number of solutions.

Table 5 Oracle SDO_NET_MEM shortest path functions (Murray, An et al. 2007).

The SDO_NET_MEM.NETWORK_MANAGER.SHORTEST_PATH function is used because the A* search implementation in Oracle is assumed[16] to be more efficient then the Dijkstra implementation in calculating the shortest routes (Murray, An et al. 2007). The following psql statement is used to calculate the shortest route:

```
res_num := sdo_net_mem.network_manager.Shortest_path('PEDMODEL', fromnode,
tonode, NULL);
```

The last parameter is NULL because an optional constraint functionality in Oracle is not used. The resulting path is identified with a new number which is returned by this function. After a save of the network this number is the path_id in the route table.

6.2.2 Storing optimal routes

The table route and routeelement are created with create_sdo_network procedure described in paragraph 6.1.1. These tables are used for storing the optimal calculated route (detailed table descriptions for table route and routeelement are found in appendix M). The route table contains an entry for every found route. As described in the previous paragraph a new path_id is generated by the shortest path calculation. This path id indentifies the route. Fields in the route table are the name of the route as generated by the test program, the origin NodeSurface id, the destination NodeSurface id. The path_type stores the speedtype and the path duration is stored in the cost field. The geometry fields are not used in the route table because Oracle supports only 2 D similar as in the link and node tables. The routeelement table stores the links visited by the path. It contains only the fields to indentify the path, the link and sequence number for every LinkSurface in the path.

[16] Comparing of search algorithms is considered outside the scope of this research.

The route and routeelement table are filled with the following PL/SQL statement in the after the route calculation:

```
sdo_net_mem.path.Set_name('PEDMODEL', res_num, pathname);
sdo_net_mem.path.Set_type('PEDMODEL', res_num, speedtype);
sdo_net_mem.network.Add_path('PEDMODEL', res_num);
sdo_net_mem.network_manager.Write_network('PEDMODEL');
```

The Oracle views Testresults and Pathview are developed during this research to inspect the results in the route and routeelement table. The SQL statements to create these views are given in Appendix O.

The Oracle view Testresults gives an overview of the route including a list of NodeSurfaces and the total duration of the route the Table 6 gives sample contents of this view.

PATH_ID	PATH_NAME	LINKSURFACELIST	Duration
1	#BoardingPosition_positionOnRailPlatform 0 to #BoardingPosition_positionAtBusStop 01-3 WheelChair Slow	240 219 185 225 61 80 70 82 48 11 258 71 77 89 85 105 106 98 101 86 97 216	570

Table 6 Example of test results output

The Oracle view Pathview gives in detail the visited links of the route. This view is also used for visualisation. Table 7 is showing the details of the links. In this Table 7 the pathnames and geometry of the start_NodeSurface is not shown. This view is also used to export the path to a KML file for visualisation with FME therefore the pathname and geometry are available in this view.
.

SEQ_NO	LINK_ID	LINK_NAME	Distance	Timeduration
1	240	#BoardingPosition_positionOnRailPlatform 0 -> #Quay 25	7	15
2	219	#Quay 25 -> #Quay 14	9	19
3	185	#Quay 14 -> #Quay 27	4	8
4	225	#Quay 27 -> #Quay_Entrance 6	10	22
5	61	#Quay_Entrance 6 -> #AccessSpace 3	3	7
6	80	#AccessSpace 3 -> #CheckPoint_lift 0 (Down)	7	60
7	70	#CheckPoint_lift 0 -> #AccessSpace 4 (Down)	4	8
8	82	#AccessSpace 4 -> #Quay_Entrance 2	7	15
9	48	#Quay_Entrance 2 -> #AccessSpace_Underpass 0	3	7
10	11	#AccessSpace_Underpass 0 -> #CheckPoint_ramp 0 (Up)	20	48
11	258	#CheckPoint_ramp 0 -> #AccessSpace 0 (Up)	35	74
12	71	#AccessSpace 0 -> #AccessSpace 2	8	17
13	77	#AccessSpace 2 -> #AccessSpace 7	12	26
14	89	#AccessSpace 7 -> #AccessSpace 5	26	54
15	85	#AccessSpace 5 -> #AccessSpace 12	18	38
16	105	#AccessSpace 12 -> #AccessSpace 13	3	6
17	106	#AccessSpace 13 -> #AccessSpace 10	7	15
18	98	#AccessSpace 10 -> #AccessSpace 11	4	9
19	101	#AccessSpace 11 -> #AccessSpace 6	6	13
20	86	#AccessSpace 6 -> #AccessSpace 9	9	19
21	97	#AccessSpace 9 -> #Quay 24	38	79
22	216	#Quay 24 -> #BoardingPosition_positionAtBusStop 0	5	11

Table 7 Example of pathview output (not showing the pathname and the link geometry

6.2.3 Visualisation of optimal routes

Visualisation is done in Google Earth. For this visualisation a KML file is generated. It is possible to create the basis of a KML file with Oracle PL/SQL in the following way:

```
select  ' <Placemark><name>seqno: ' || T1.seq_no || ' nodeid: '|| T3.node_id
|| ' nodename: '|| T3.node_name || '</name>' ||
          xmltype (SDO_UTIL.TO_KMLGEOMETRY(T2.geometry3d)) ||
'<description>' || T2.start_node_id || ' to '|| + T2.end_node_id  ||  '
</description> <gx:balloonVisibility>1</gx:balloonVisibility></Placemark>'
         FROM         ROUTEELEMENT t1, LINKSURFACE t2, NODESURFACE t3
         WHERE     t1.LINK_ID = t2.LINK_ID AND T3.node_id = T2.end_node_id
          order by T1.path_id,  T1.seq_no;
```

The output of this SQL statement was then copied to the body of an empty KML file. This KML file can then be opened in Google earth. It was found that in Google earth that it is not possible to show surfaces below the ground level. Therefore the model is raised 10 meters above ground level, for visualisation only, to make it possible to show the NodeSurfaces under the ground (e.g. the underpass). It has a side effect that all NodeSurfaces are "flying" above earth in the visualisation. For a traveller this is a suboptimal situation but to validate the model usable[17]. Furthermore it was find out during the research that this way of producing KML files is time consuming and a source for errors, therefore the data conversion tool FME was used. In Figure 38 is shown how the conversion is modelled in the FME workbench.

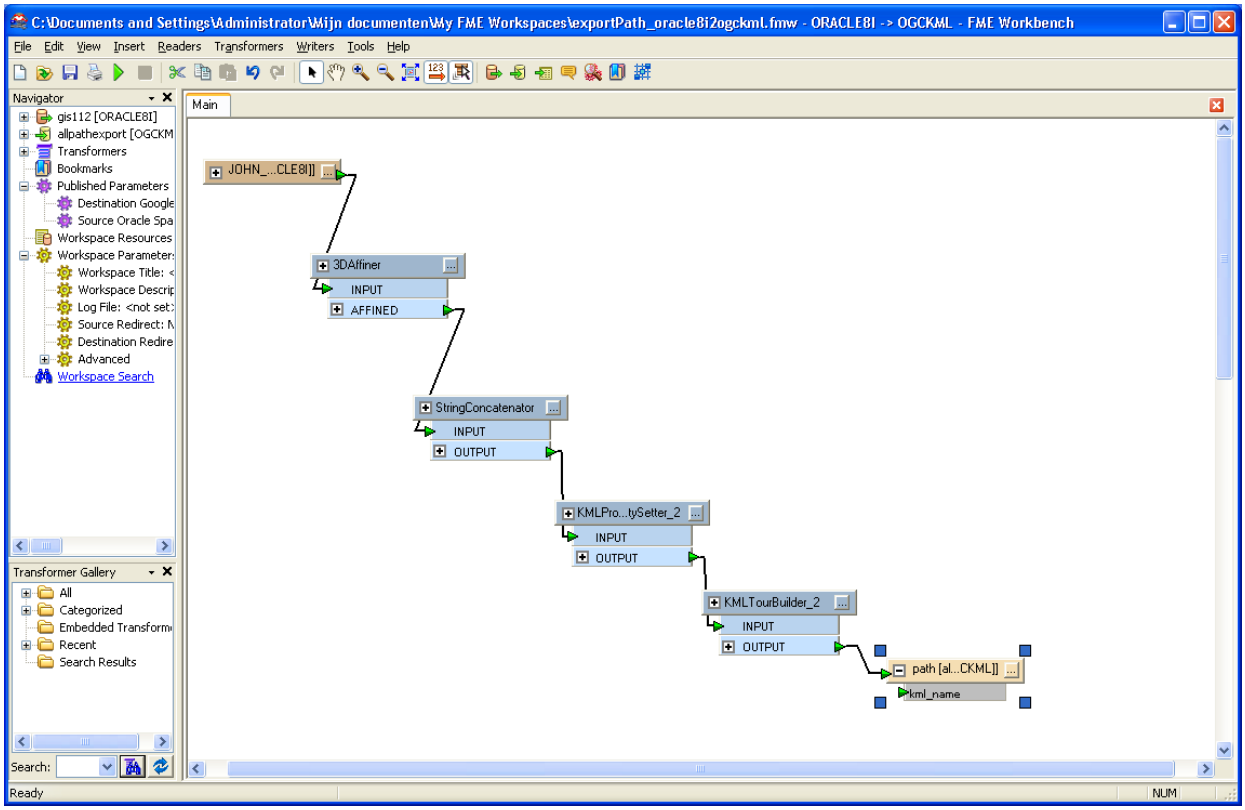


Figure 38 FME workbench used to convert Oracle optimal routes a KML file

[17] Optimizing visualisation for traveller is considered outside the scope of this research

In FME the following actions are done:

1) Connect to oracle and the pathview is loaded
2) With the function "3D affiner" 10 meters to all Z coordinates (height) is added
3) A string is created naming the NodeSurfaces in the LinkSurfaces
4) The result are converted to a KML file
5) A tour[18] is created for all LinkSurfaces from all paths found in sequence in the pathview.
6) The KML file is written to disk

The generated KML file can be opened in Google earth to inspect all details of the generated routes. Figure 39 illustrates how visualisation in Google Earth takes place. The route can shown by the using the tour feature (Google 2009a). The tour player tool, looking video player buttons, to start a tour is showed in the left lower part of the image in the Google earth view. Starting a tour simulates a flight following the links of the paths in sequence.

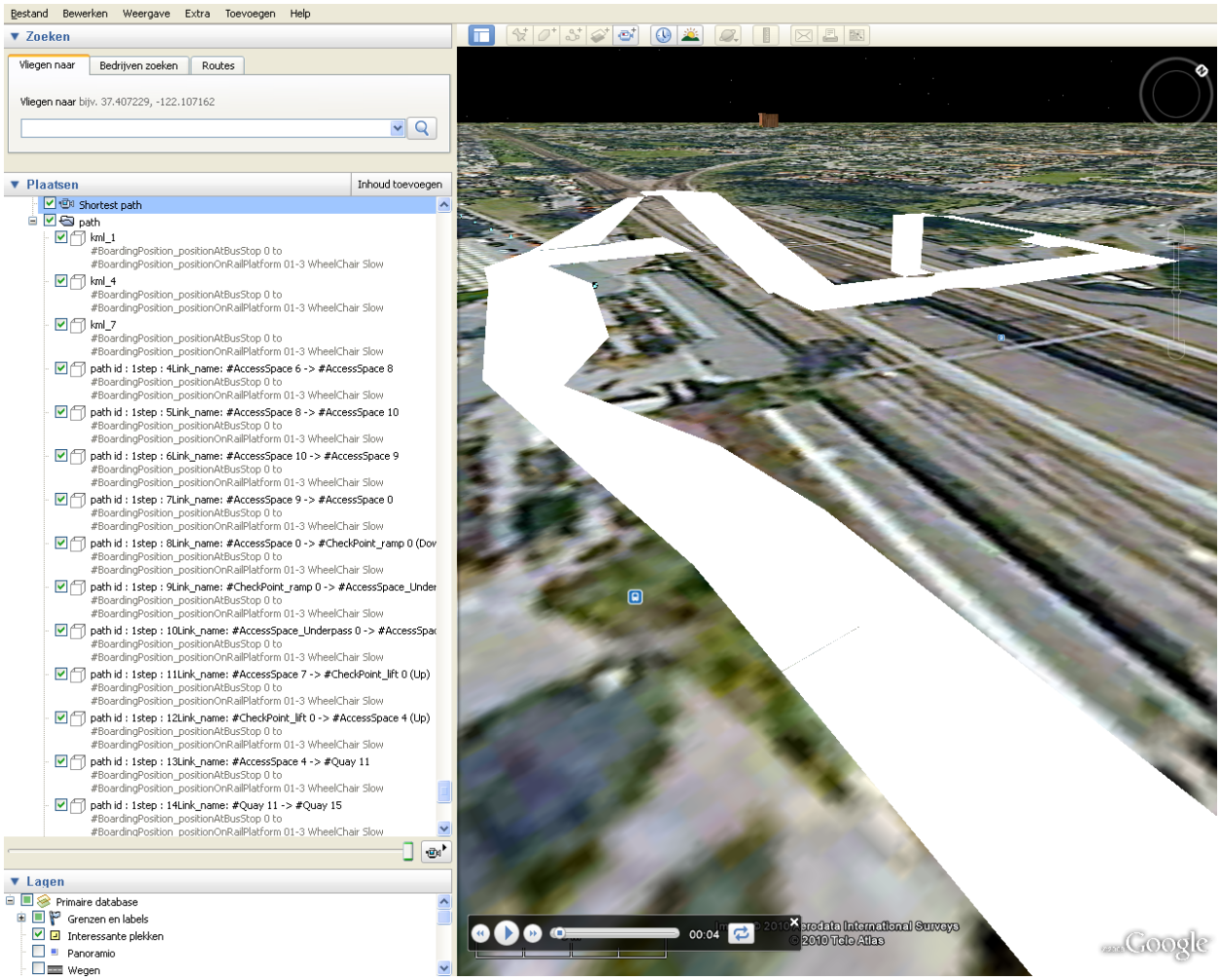


Figure 39 Viewing a route in Google earth as tour

[18] A tour is a KML extension to visualise a route by a controlled flight in a Google earth session (Google 2009a).

7 ANALYSIS OF RESULTS

As part of the research approach scenes and scenarios for pedestrian route were developed to evaluate the prototype. In chapter 4 (paragraph 4.4 on page 31) these scenes and scenarios for evaluation of a new geo-data model were given. These scenarios and scenes are used to do experiments with the developed prototype. This chapter first focuses on the experiments to evaluate unrestricted routing with the prototype in section 7.1. After this unrestricted routing the experiences with the prototype, using the traveller profile with speed and accessibility capabilities in routing are described in 7.2. After this functional evaluation of the prototype the experiences with Oracle 11g and additional software, used in this research, are described in section 7.3.

7.1 Evaluating the prototype on routing in scene space

The use case "Locating the entrance (s) of a public transport place in relation to the surroundings"` was evaluated. For this model the origin "Forum" and destination bus stop "droevendaalse steeg direction Ede" (Figure 31) was selected. In Google earth the NodeSurface where drawn between the origin and destination. These NodeSurfaces where exported as KML files. This model did work as expected, shortest routes where found taking the shortest direction over the walk able areas near a roundabout.

After this experiment a location Ede-Wageningen (Zuid) was modelled (Figure 40). This location was used to validate the scenarios for a multimodal transport place.

The following scenario was used for validation of the prototype:

- Navigate "inside" a multi modal transport place, considering steps, stairs, ramps and elevators.
- Change modality bus to train

The experiments started focussing on the distance of an optimal path for unrestricted travellers. This analysis was done first on a simple model containing "walk around" NodeSurfaces and is described in paragraph 7.1.1. Later this was done with a more advanced model having "hidden paths" modelled, this experiment is described in 7.1.2.
With the hidden path model the experiments continued. Pedestrian routing with the traveller profiles for speed for unrestricted travellers were analysed (paragraph 7.2.1). After this analysis the focus was on analysing routing with traveller profiles containing mobility needs (paragraph 7.2.2). Each single combinations of traveller speed and mobility need was analysed. At last is evaluated how the direction of optimal route (change train to bus instead of bus to train) is influencing the outcomes (paragraph 7.2.3).

The prototype is validated by let the prototype finding optimal routes between the bus stop Ede Zuid and the nearest entrance from a train in the direction of Utrecht. The tested location is not very complex. For a traveller with local knowledge and no restrictions an optimal route can be straightforward found. This shortest route, which is usually taken by unrestricted travellers between the bus stop and the boarding position, is shown in Figure 40. Please note that the point locations supplied by Google (the pictograms in blue') use in Google Earth to announce the train station and bus stop are not the entrance and egress positions from the vehicles used by the prototype. Furthermore the walk is using the south entrance of the station, taking the stairs to the under passage then right stairs up to the train platform (IFOPT:Quay). For people not able to use stairs there is a ramp down to the under passage at the left hand when entering the south entrance. Furthermore there is a lift in the underpass to the platform. This lift is located opposite of the stairs to the platform in the underpass (left). This location is also shown in Figure 32 on page 53. The length of the route (2D) is estimated with the Google lineal tool. Figure 40 illustrate this route and the use of the Google lineal tool. The accuracy of the distance is given in centimetres is questionable compared to the level of detail of the image. The usability of the Google lineal tool was tested by measuring a object with know length (50 meter swimming pool) as shown in appendix P

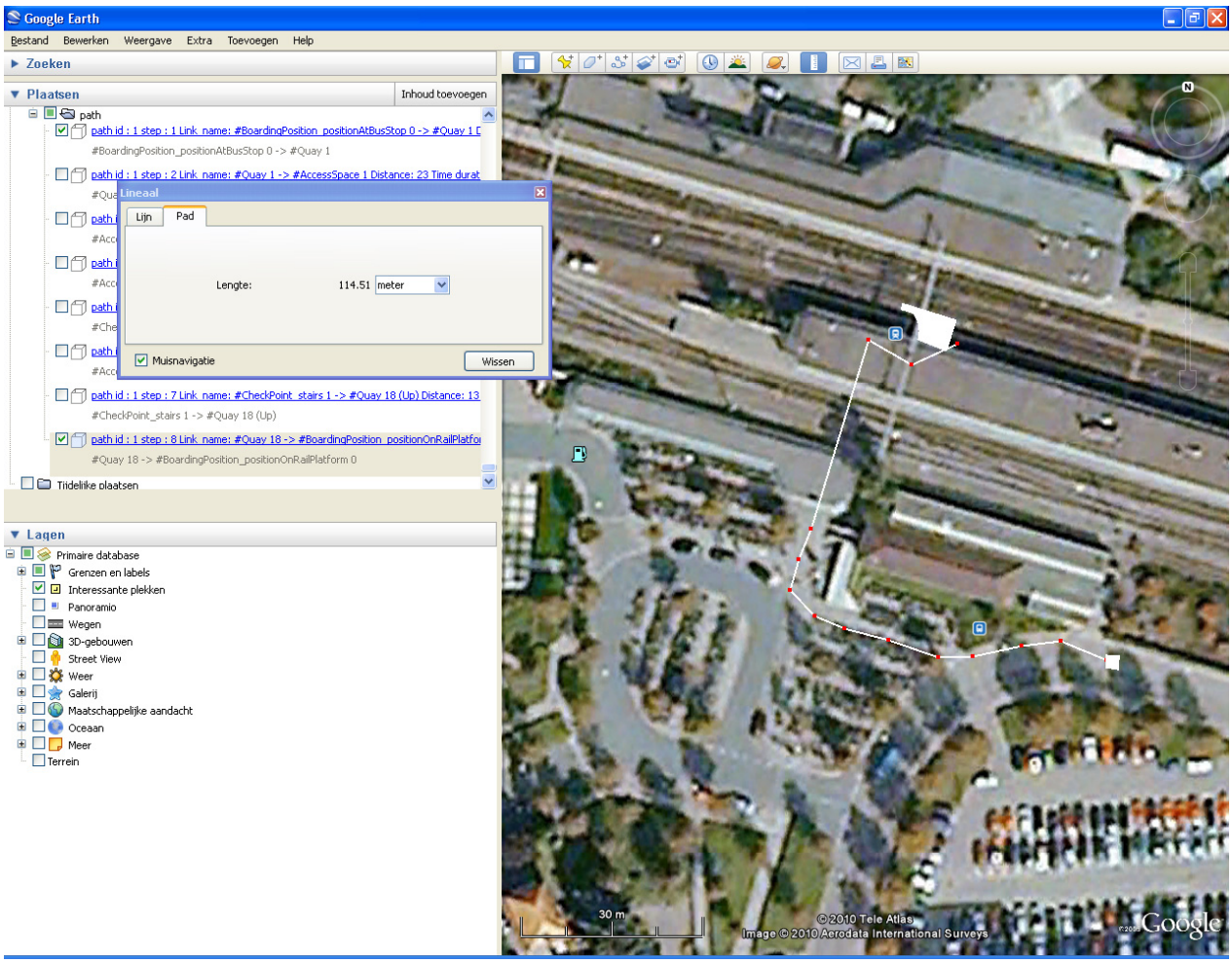


Figure 40 Walking route between bus stop and entrance of a train.

7.1.1 Optimal route with "walk around" NodeSurfaces

Figure 41 shows that the expected route is found by the prototype. This figure shows that prototype does 3D support. The underpass and the platforms not direct connected, it only connects the underpass with the stairs (at the entrance of the station) and stairs to the train platform. In a 2D model (e.g. map) a direct connection (overlap) exists between the train platform and the underpass. In Figure 41 the route follows the route line in Figure 40 and is not using the lift or ramp. This is correct because this would take longer in case of an unrestricted traveller.

An unexpected behaviour of the model is that it connects some NodeSurfaces by the corners only. This is shown by the connection of the underpass and the stairs (see Figure 41 and the annotation "Underpass stairs and train platform (Quay) not properly connected"). A modelled hidden platform entrance as described in paragraph 5.5, is creating a hole in the route, it should be part of the route. The route is using the "3D touch" relation to connect the underground surface and the stairs. The origin of this shortcoming in the prototype is the usage of the Oracle function SDO_ANYINTERACT to find connecting NodeSurfaces in 3D space (mentioned on page 55 footnote 13). In 3D space Oracle does not support other spatial relations then "ANYINTERACT" between two surfaces in 3D. In 2D SDO_RELATE can be used with various masks to test for the different kind of spatial relations. This is not supported in 3D Space, only the ANYINTERACT is supported by Oracle 11.2 (Murray 2010).

The connection as shown in Figure 41 between the underpass and the stairs to the platform is an invalid LinkSurface according the conceptual model. It connects 2 point (2 touched corners) between two surfaces which is not allowed as described in paragraph 5.3 and showed in Figure 18.

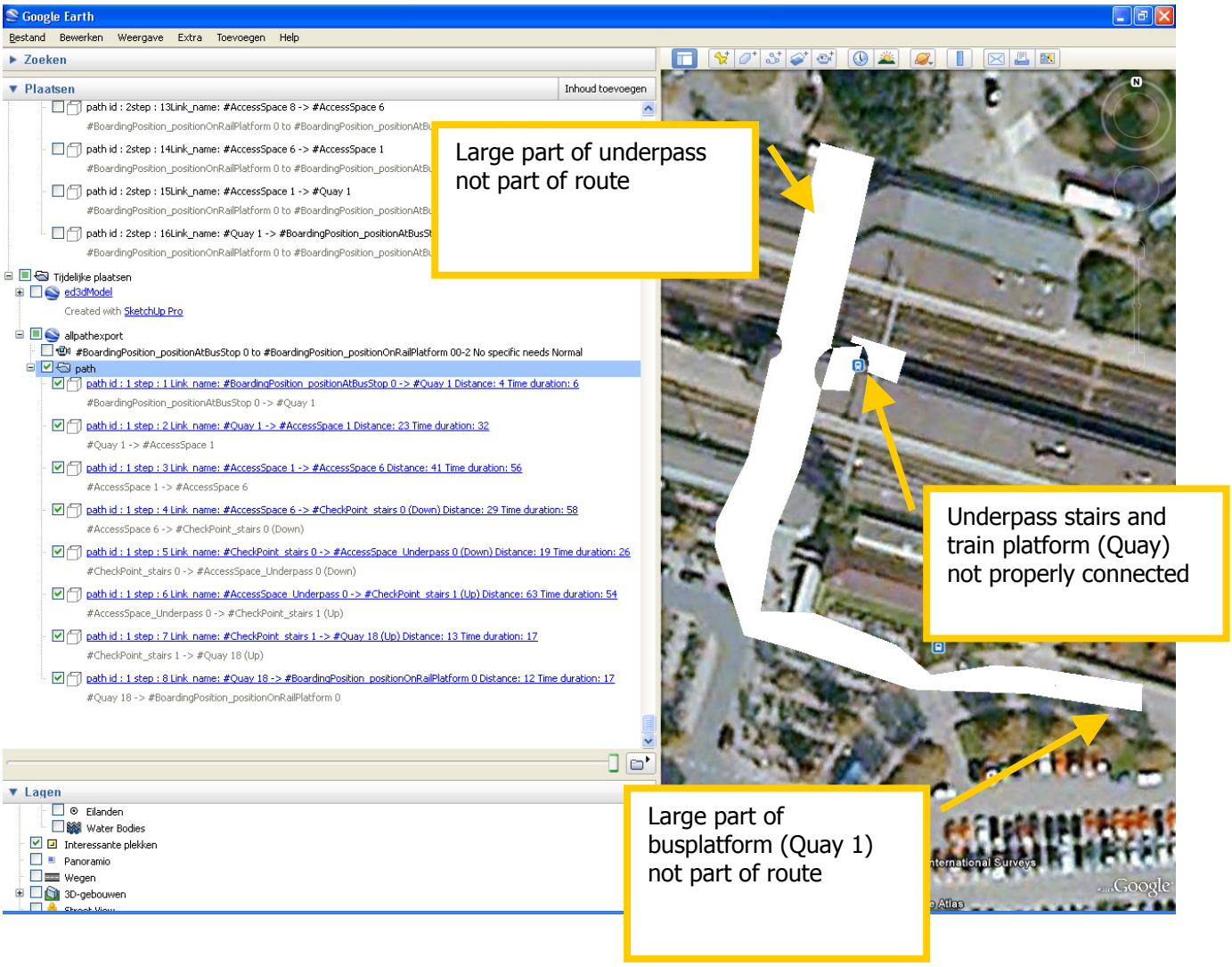


Figure 41 Optimal walk route between bus stop and entrance of a train found by the prototype (using stairs down to underpass and the stairs up to platform

A shortcoming of the prototype is that the calculated length of the route is much longer 204 meter compared to the rounded (114,51 meter) value of 115 meter calculated with the Google line tool. Table 8 is showing lengths for the passed node surfaces. Differences can be explained because of two types of shortcomings of this prototype:

1) Especially the walking distance of the underpass and the busplatform (IFOPT: Quay) is longer because only half the surface is used to connect make the connections between the node surfaces.
2) The used algorithm to estimate the distance from the link over the NodeSurfaces estimated by taking half the perimeter of the NodeSurface as described in 6.1.3 does not give exact the shortest distance for the connection.

The first limitation was studied by modelling the studied area in smaller NodeSurfaces, drawing logical walk routes on surfaces. The use of (smaller) surfaces modelling logical walk routes "hidden paths" will estimated better distances. Half the perimeter of a NodeSurface is a more accurate estimate for the distance of a connection when the NodeSurface (hidden path) is narrow compared to the estimate of a complete ("walk around") surface.

Link description	Horizontal distance in meters from prototype	Horizontal distance in meters measured with Google line tool
#BoardingPosition_positionAtBusStop 0 -> #Quay 1	4	3
#Quay 1 -> #AccessSpace 1	23	7
#AccessSpace 1 -> #AccessSpace 6	41	19
#AccessSpace 6 -> #CheckPoint_stairs 0 (Down)	29	25
#CheckPoint_stairs 0 -> #AccessSpace_Underpass 0 (Down)	19	12
#AccessSpace_Underpass 0 -> #CheckPoint_stairs 1 (Up)	63	22
#CheckPoint_stairs 1 -> #Quay 18 (Up)	13	9
#Quay 18 -> #BoardingPosition_positionOnRailPlatform 0	12	13
Total distance	204	110

Table 8 Distance for every NodeSurface passed in optimal walk route between bus stop and entrance of a train

7.1.2 Optimal route with "hidden path" NodeSurfaces

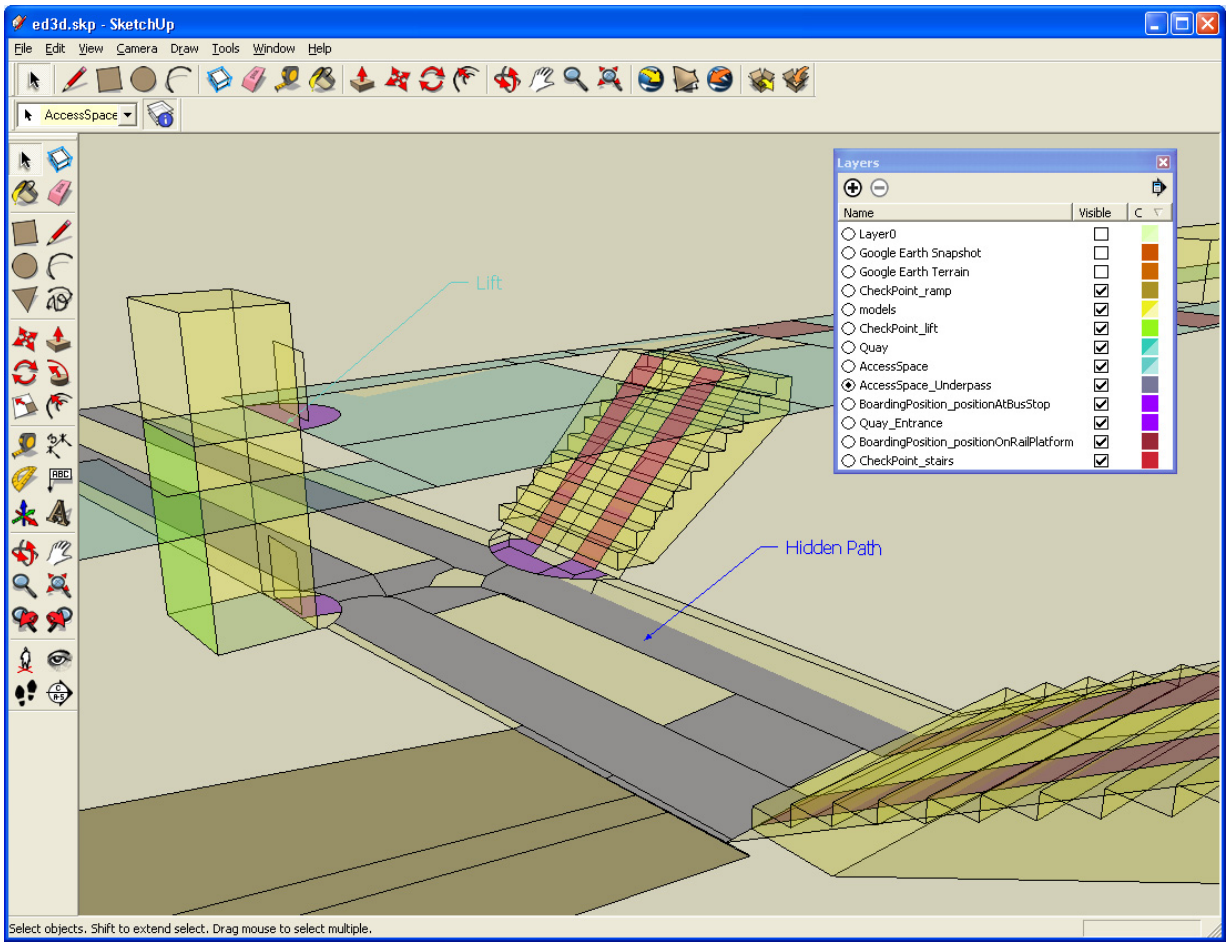


Figure 42 NodeSurfaces modelled as hidden paths in the underpass

To further evaluate the prototype NodeSurfaces modelling detailed walk routes on surfaces (hidden paths) where added to the model. The hidden paths in the underpass are show in Figure 42. The hidden paths are avoiding corners. The corners are modelled as non pedestrian areas and only a part of the original NodeSurfaces are used for pedestrian access. The side effect that the algorithm can connect NodeSurfaces by corners does not occur in this situation.. In the studied model it was possible

to model hidden paths in such way that no paths are touching each other with the corners only. The distances found, when routing over the hidden paths, is showed in Table 9. Figure 43 shows the optimal path. In this figure can be seen that the limitations showed Figure 41 in are not occurring when modelling hidden paths in this way. Table 9 shows that distances calculated by the prototype using hidden paths are more in line with the distances found with the Google line tool.

Link description	**Horizontal distance in meters from prototype**	**Horizontal distance in meters measured with Google line tool**
#BoardingPosition_positionAtBusStop 0 -> #Quay 25	4	3
#Quay 25 -> #AccessSpace 16	5	5
#AccessSpace 16 -> #AccessSpace 10	38	42
#AccessSpace 10 -> #AccessSpace 11	9	8
#AccessSpace 11 -> #CheckPoint_stairs 0 (Down)	14	14
#CheckPoint_stairs 0 -> #AccessSpace_Underpass 8 (Down)	12	12
#AccessSpace_Underpass 8 -> #Quay_Entrance 1	18	17
#Quay_Entrance 1 -> #Quay_Entrance 2	4	2
#Quay_Entrance 2 -> #CheckPoint_stairs 2 (Up)	2	1
#CheckPoint_stairs 2 -> #Quay 19 (Up)	7	5
#Quay 19 -> #Quay 22	2	2
#Quay 22 -> #BoardingPosition_positionOnRailPlatform 0	6	4
Total	121	115

Table 9 Distance for every NodeSurface passed in optimal walk route between bus stop and entrance of a train using hidden paths

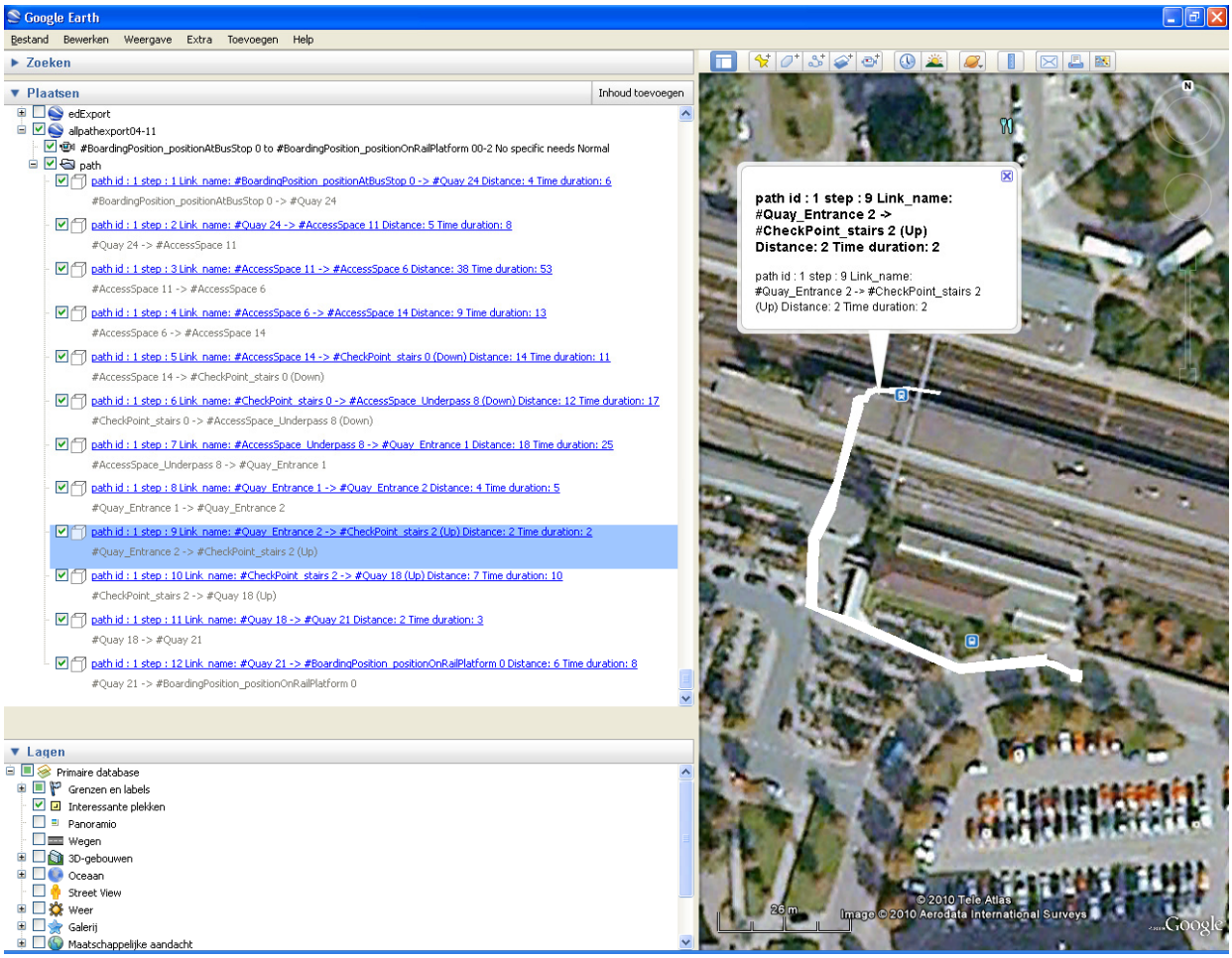


Figure 43 Route 1: Optimal walk route between bus stop and entrance of a train found by the prototype using hidden paths for unrestricted travellers (using stairs down to underpass and the stairs up to quay)

7.2 Evaluating the prototype pedestrian routing capabilities

Based on the finding described in the previous session the experiments taking the traveller profile into account continued with the hidden path model. First pedestrian routing with the traveller profiles for speed for unrestricted travellers was first analysed Second the focus was on analysing routing with traveller profiles containing mobility needs (paragraph 7.2.2). Third all single combinations of traveller speed and mobility needs were analysed. After that is evaluated how the direction of optimal route (change train to bus instead of bus to train) was influencing the outcomes (paragraph 7.2.3).

7.2.1 Analysis of the speedtype in travellerprofiles for unrestricted travellers

As described in paragraph 6.1.3 the prototype first add durations to LinkSurfaces according to speed profiles before it calculates the optimal route. The scenario in the previous experiment was also evaluated taking the speed profiles into account. No different routes where generated by the prototype for the different access profiles. This could be the case when e.g. a lift or an escalator would be faster then slow walking a staircase. The duration results per LinkSurface are shown in Table 10. All LinkSurfaces in the route found where objects with the assumption[20] that fast travellers can increase there walk speed with 50 percent on these objects and slow travellers are using a speed with 50

[20] Research into these walk speed and lift profile values is considered outside the scope of this research.

percent less of the average traveller on these objects. Therefore the total durations differs also 50 percent of the normal duration. The results are looking promising assuming more than 4 minutes for a interchange to support slow pedestrians and 1.3 minute for a fast pedestrian seems reasonable to the author knowing the actual situation at the studied location, but as mentioned before comparing these walk durations with field studies results is needed to qualify these values.

Id	Link Description	Distance[21]	Fast[22]	Normal[22]	Slow[22]
116	#BoardingPosition_positionAtBusStop 0 -> #Quay 24	4	3	6	9
215	#Quay 24 -> #AccessSpace 9	5	4	8	11
93	#AccessSpace 9 -> #AccessSpace 6	38	26	53	79
87	#AccessSpace 6 -> #AccessSpace 12	9	6	13	19
103	#AccessSpace 12 -> #AccessSpace 15	6	4	9	13
109	#AccessSpace 15 -> #CheckPoint_stairs 0 (Down)	10	4	8	12
259	#CheckPoint_stairs 0 -> #AccessSpace_Underpass 5 (Down)	12	9	17	26
25	#AccessSpace_Underpass 5 -> #Quay_Entrance 0	18	13	25	38
41	#Quay_Entrance 0 -> #Quay_Entrance 1	4	3	5	8
45	#Quay_Entrance 1 -> #CheckPoint_stairs 2 (Up)	2	1	2	3
2	#CheckPoint_stairs 2 -> #Quay 18 (Up)	7	5	10	14
196	#Quay 18 -> #Quay 21	2	2	3	5
205	#Quay 21 -> #BoardingPosition_positionOnRailPlatform 0	6	4	8	13
	Path duration	123	83	167	249

Table 10 Prototype results for walk durations for unrestricted travellers

7.2.2 Analysis of the mobility needs in traveller profiles.

In the previous paragraph route calculation was done for a traveller profile with no specific needs. To evaluate the prototype route calculation for the same situation using hidden paths was done for all the other accessibility profiles considering "normal" speed first. Four different solutions where found by the prototype consisting of three different routes and one not possible connection for the access profile with "No down stairs/ramps". Table 11 is showing these 3 routes and the not possible connection found. The routes for "averse to lifts" and "escalator free" was exact the same as in the previous paragraph (route 1). This result is caused because no escalator exists in the studied area and the route found in the previous paragraph does not make use of lifts. This route is the fastest for the "normal speed" traveller, slower is the (route 2) found when this traveller is restricted with "Luggage medium". Figure 44 show this second route. Interesting in this route in the fact that this route is taking the lift and the route for this traveller starts walking on down the left side of the stairs to find the fastest way to this lift. This left side walking happens because directions are taken into account on the hidden paths only for up and downwards accessibility as described in paragraph 6.1.2. No restrictions in hidden paths are defined for walking left side of the original NodeSurface where the hidden paths are part of. Similar results in the prototype are seen for the third route found for the restricted traveller profiles having step free, wheelchair or heavy luggage needs (route 3). As seen in Figure 45 the ramp is taken and the shortest hidden path (the one left seen from the travelled direction) is taken. Except for left side walking in the underpass this route is make sense for travellers with step free or wheelchair access. If this route is better then route 2 is for heavy luggage needs is uncertain, if the luggage has wheels and can be towed, this routes might be optimal, if you carry heavy luggage, you might prefer a shorter walk along stairs instead of a long walk using a ramp.

[21] Distance in meter (rounded values)
[22] In seconds (rounded values) for the in paragraph 5.7.2 described speed profiles

Access profile	Sequence of NodeSurfaces (id's) in route	Route	Total duration Normal[22]
No specific needs	116 215 93 87 103 109 259 25 41 45 2 196 205	1	167
WheelChair	116 215 93 87 102 99 107 104 84 88 76 72 257 10 50 83 69 78 63 229 184 220	3	390
Step Free	116 215 93 87 102 99 107 104 84 88 76 72 257 10 50 83 69 78 63 229 184 220	3	390
Averse to lifts	116 215 93 87 103 109 259 25 41 45 2 196 205	1	167
Escalator free	116 215 93 87 103 109 259 25 41 45 2 196 205	1	167
Luggage heavy/Normal	116 215 93 87 102 99 107 104 84 88 76 72 257 10 50 83 69 78 63 229 184 220	3	390
Luggage medium/Normal	116 215 93 87 102 100 111 389 10 50 83 69 78 63 229 184 220	2	264
No down stairs/ramps		-	-

Table 11 Prototype results for "normal speed" travellers in seconds

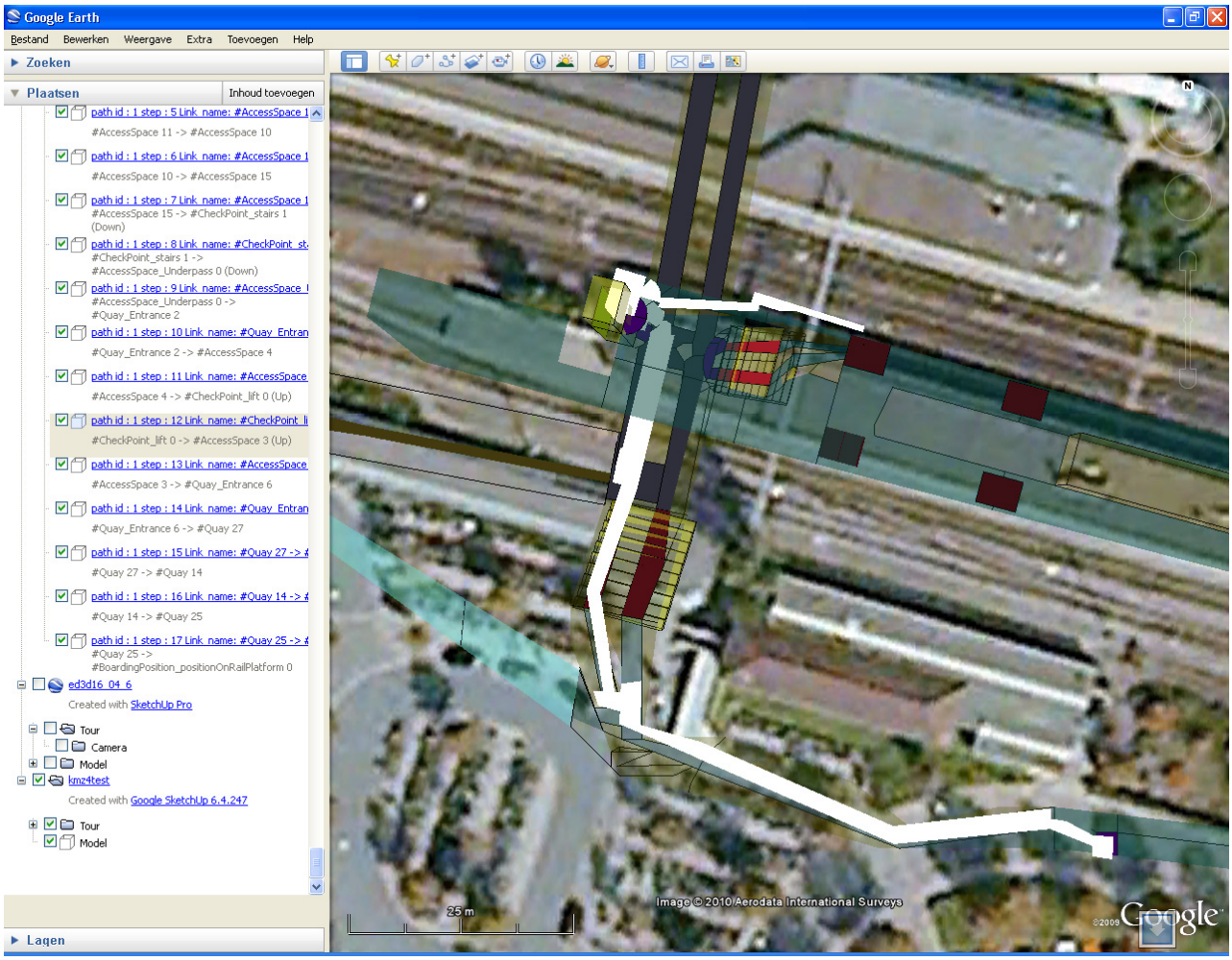


Figure 44 Route 2: Optimal walk route between bus stop and entrance of a train found by the prototype using hidden paths for travellers restricted with medium luggage (using stairs down to underpass and the lift up to Quay

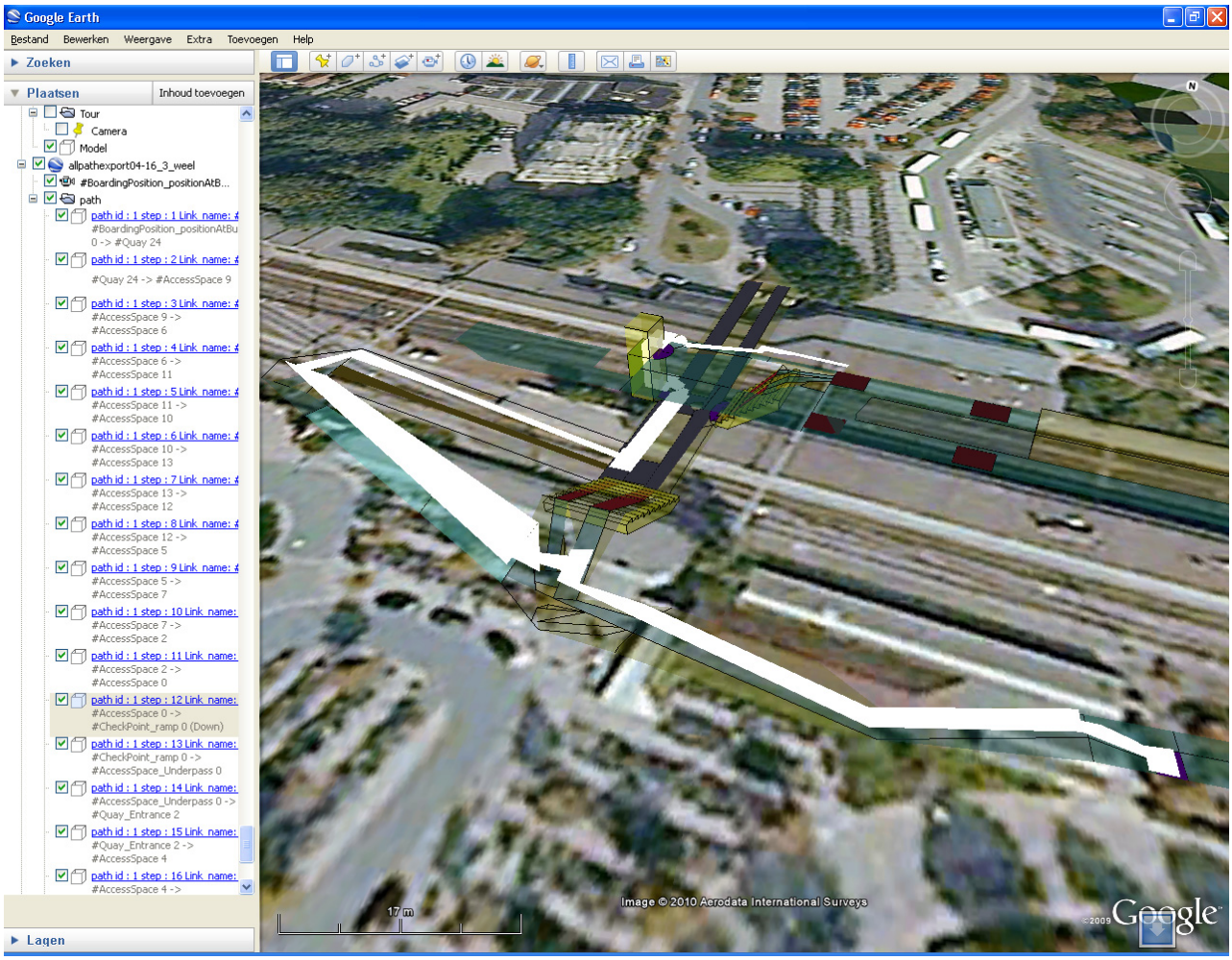


Figure 45 Route 3: Optimal walk route between bus stop and entrance of a train found by the prototype using hidden paths for travellers restricted by or step free access, heavy luggage or wheelchair (using ramp down to underpass and the lift up to Quay)

7.2.3 Analysis of the combinations of speedtype and mobility needs in both directions

The prototype is tested to find routes for all accessibility needs and the 3 speed profiles. No new routes were discovered using the speed profiles. This could be the case when e.g. a lift or an escalator would be faster then slow walking using a staircase. The results are shown in Table 12

Access profile	Route	Fast	Normal	Slow
No specific needs	1	83	167	249
WheelChair	3	225	390	555
Step Free	3	225	390	555
Averse to lifts	1	83	167	249
Escalator free	1	83	167	249
Luggage heavy/Normal	3	225	390	555
Luggage medium/Normal	2	162	264	366
No down stairs/ramps	-	-	-	-

Table 12 Prototype route duration results bus to train scenario in seconds.

As described before all LinkSurfaces in the route 1 were objects with the assumption that fast travellers can increase their walking speed with 50 percent and slow travellers are using a speed with 50 percent less of the average traveller on these objects. This is not the case for route 2 and 3. The influence of the fixed time for lift usage makes the difference in the total durations for the different speed profiles smaller for route 2 and 3.

To further test the prototype the same evaluation is done for route finding in the opposite direction (train to bus). Although the model supports different exits and entrances for public transport in this evaluation is assumed that the same NodeSurfaces are used for entrance and exit of vehicles. The results of the opposite routes are shown in Table 13. The route durations are similar in both directions. The differences are the result of different in speed for ramp up or ramp down and size differences of start_NodeSurface and end_NodeSurface. The prototype includes the distance in the duration calculation of the start_NodeSurface but no the distance for the end_NodeSurface of the pedestrian part. This because the distance is calculated out of the start_NodeSurface of a LinkSurfaces. The size of the reached destination NodeSurface is not part of the path but can be seen as start for a new path. Conceptual this means that the model does not takes into account the exact where the pedestrian is on the start node surface but it knows the maximum distance to reach the next NodeSurface in the path. At the end surface the model assumes that the pedestrian is on the end_NodeSurface on a location touching the previous NodeSurface in the path. Extending the prototype and considering the pedestrian part inside a vehicle is outside the scope of this research but in that case the end location would be in the vehicle at the seat location. To take the side of the boarding position into account in a route calculation the vehicle location would be a better end NodeSurface of a pedestrian route instead of the IFOPT boarding position taken now as end NodeSurface of a pedestrian path.

Access profile	**Route**	**Fast**[22]	**Normal**[22]	**Slow**[22]
No specific needs	1	82	162	244
WheelChair	2	230	400	570
Step Free	2	230	400	570
Averse to lifts	1	82	162	244
Escalator free	1	82	162	244
Luggage heavy/Normal	2	230	400	570
Luggage medium/Normal	3	171	341	512
No down stairs/ramps	4	161	261	362

Table 13 Prototype route duration results train to bus scenario.

7.3 Experiences with database and tools used in the prototype

The Oracle 11g database was used to create the prototype. Next to the Oracle database the following tools where used to create this prototype:

1) Google SketchUp
2) Enterprise architect
3) FME
4) SQL developer
5) Google earth
6) C# visual studio + GraphSharp

In appendix A references are made to sites describing these tools.

This section describes the findings on the Oracle usage in this research in the next paragraph. After the findings on Oracle paragraph 7.3.2 describes brief the experiences with the other tools.

7.3.1 Evaluating Oracle Spatial usage in the Prototype

In this research working with the spatial and network capabilities of Oracle 11g to analyze connectivity of 3D surfaces has been challenging. Getting the prototyping working using the Oracle PS/SQL, the spatial functions and the supplied Oracle network functions took more time than expected. Especially the way how errors and exceptions are communicated to a PL/SQL user is remarkable. Consider the helpfulness of an error message caught when using an unsupported geometry in an Oracle spatial network for a user not having access to the Java sources:

```
ORA-29532: Java call terminated by uncaught Java exception:
oracle.spatial.network.NetworkDataException:
java.lang.NullPointerException
ORA-06512: at "MDSYS.SDO_NETWORK_MANAGER_I", line 284
ORA-06512: at "MDSYS.SDO_NETWORK_MANAGER_I", line 114
ORA-06512: at line 15
29532. 00000 -  "Java call terminated by uncaught Java exception: %s"
*Cause:    A Java exception or error was signaled and could not be
           resolved by the Java code.
*Action:   Modify Java code, if this behavior is not intended.
```

Furthermore the limitations of the Oracle network analyse functions only to support specific 2D geometries are not specified in the documentation. It also takes some time understand the impact of the 3D limitations of the various other Oracle SDO functions. Oracle's logic that when using 3D objects in functions, instead of not functioning, are assuming that a 2D result is useful is questionable. As stated in the manuals (Murray 2010) that for some functions "the height information is ignored when dealing with geodetic data". For distance computations with 3D geometries it works as follows:

- If the data is geodetic (geographic 3D), the distance computations are done on the geodetic surface.
- If the data is non-geodetic (projected or local), the distance computations are valid only if the unit of measure is the same for all three dimensions.

The limitation of the Oracle network capabilities not supporting 3D route calculations did influence the prototype construction process significant. The SDO_geometry fields which can be used for line topology can not be used to store surfaces. To store the geometry of the NodeSurfaces an extra field was added to the node table. The standard aggregate function in the Oracle network software to aggregate link geometries in a final route can not be used for 3D surfaces. In the model only the logical aspects of the network model are used and SQL joins are used to extract the geometry of the route.

In Oracle 11g Boolean java functions can be added to nodes in a network which are evaluated during shortest path search. These functions, called constrains, can be used to determine if a node or link is valid to take during the shortest path search (Murray 2010). For a non experience Java programmer, lacking solid Java skills and access to Oracle Java library sources it was not possible in the timeframe of this research to make a prototype using constraints for optimal route planning. The method used in this prototype to update the durations in the LinkSurface table and loading the network into memory for each route calculation is not using computer resources optimal. A solution using constraints is more efficient. Studying the capabilities to analyse large networks with the load on demand implementation is not done in this research but are useful when modelling large networks. Research on the suitability of other new features like, the hierarchical shortest path, K-shortest path, a network buffer feature to derive the zone of influence with coverage and cost information and a "minimum cost polygon" to allow for the association of e.g. NodeSurfaces in a defined region are subject to more research.

7.3.2 Evaluating other tools used for the prototype.

Enterprise Architect has been a good tool when defining use cases, the conceptual and implementation model for this pedestrian model. The automatic creation of DLL was challenging but useful. It was challenging to integrate the network creation function supplied by Oracle and the database engineering functions of Enterprise Architect. The network creation function was used because pattern of an Oracle spatial network is fixed and the network creation function is using a high abstraction level for generating such network model in Oracle. Furthermore this Oracle creation functions is also creating all necessary metadata for the network immediately. Therefore first the network model is generated by the Oracle functions and reengineered to Enterprise Architect. All other tables were added afterwards with the database engineering functions of Enterprise Architect. A different approach is to create all tables with Enterprise Architect and building all network tables first and adding the logical network functionality with PSQL statements afterwards.

Google SketchUp 6 has been a useful tool to draw 3D geometries. It is user-friendly; the ability to import existing 3D models as visual clues in a drawing has been useful. Using layers to distinguish the various types of NodeSurfaces objects has been useful. Adding specific attribute values to NodeSurfaces and exporting them in KML files is field for further study. Also was found out during this research that the KML format generated by Google SketchUp 7 Pro is significant different than the format produced by Google SketchUp 6 using KML extensions to include a COLLADA files (with extension 'DAE") in compressed KML files. FME is a useful tool to transform the exported compressed KML files (kmz) models from Google SketchUp 6 to Oracle spatial.

The FME workbench and it conversion functions made it also easy to export the optimal routes to KML to analyse in Google Earth. Google Earth made it easy to visualize and analysing the results. 3D is not optimal handled in Google Earth. Underground features can standard not be viewed and 3D GIS functions are limited. The limitation that SQL developer can not visualise 3D geometric objects makes it difficult to analyse SQL queries with returning 3D geometric objects. The functionality to parse KML files made with C# in Visual Studio and the usage of GraphSharp to display networks was only from minor importance in this research. It was good to gain practical knowledge by using this variety of applications to construct a spatial database. One integrated 3D application (Suite) to use in combination with a spatial database would be useful for this kind of research.

8 CONCLUSIONS AND OUTLOOK

The intention of this research was to address the lack of current public multimodal transport planners not using the geometry of the walking part of a journey in public transport advices. 3D walking route advices for the walking parts of the public transport journey are needed. To come to an effective and well-structured geo-data model to support 3D walking route advices for the walking parts of public transport journey a research strategy was developed containing a extensive literature study, a conceptual design and an experiment. This strategy has been fruitful. The extensive literature study to gain knowledge on the complex field of the geo data models to support walking routing for public transport advices has let to the findings that no such models in this specific domain are described in literature. No existing geo-data models described are covering this complex domain considering fundamental the relations between 3D spaces, public transport, pedestrian routing, 3D visualisation and travellers and individual needs. That no such models are found in literature is an omission, therefore this final chapter starts with the question "What is essential for a 3D geo-data model to support pedestrian routing in multimodal public transport travel advices?" After this subject the final conclusion and the achievements of this research are written. As final part of this thesis an outlook is made on potential improvements in geo data models and geo data processes in this domain to support 3D walking route advices for the walking parts of the public transport journeys.

8.1 What is essential for a 3D geo-data model to support pedestrian routing in multimodal public transport travel advices?

When defining the scope of this research no requirements for a geo-data model to support this 3D pedestrian routing for the walking parts of public transport journey were know and therefore the first research questions on requirements for a geo-data model was formulated:

1. Which requirements can be defined to a geo-data model supporting access, interchange and egress walk-routing in multimodal travel advices?

After study existing models as described in chapter 3 requirements where defined for a geo-data model supporting access, interchange and egress walk-routing in multimodal travel advices. A differentiation between functional and non-functional requirements was made. Functional requirements were derived from existing public transport models found in literature. Non functional requirements were defined as approach to create a well structured geo-data model implementable in a spatial database.

Use cases (Traveller scenarios) where defined to validate the functional requirements. These use cases were used to validate the prototype as described in the previous chapter. The use cases related to pedestrian routing to entrance and exits of public transport locations and detailed pedestrian route inside a public transport locations were evaluated and supported by the conceptual model. Use cases to connect the public transport part of a journey to the pedestrian part where only validated to reach the boarding and egress position. As mentioned in paragraph 7.2.3 modelling pedestrian routing in the vehicle would improve this connectivity to the vehicle part of a public transport advice model. The conceptual model is a good basis for pedestrian routing; the use case for detailed modelling of pedestrian areas inside public transport location is supported. The capability to support logical routes on surfaces enables this. Important is that the use cases were focusing on accessibility and duration of pedestrian routing. Some topics of pedestrian routing are not included in the model. In the use cases travellers needing auxiliary services like shops are mentioned. This use case is not explicit supported by the conceptual model. Signs were mentioned as important in paragraph 3.3.2 and could be added to LinkSurfaces in the geo data model. Time based access granting as described in 3.6.2, e.g. an entrance is only open till 22.00 hours, could also be added to the model.

Non functional requirements resulted in the object modelling approach using UML to design the model. Tools were used to create a model based implementation in a spatial database automatically derived from this UML design. Standards were important in this research. The IFOPT (IFOPT 2007) standard was the most important for the semantic to describe the surfaces and other objects relevant

for travellers. The OGC standards were used for were used to exchange, store and visualize the geometric data of the locations studied in the prototype. The Simple Feature Implementation Specification for SQL (OGC 1999) is used for storage of the geometric data in the conceptual model. The Keyhole Markup Language (KML) (OGC 2008) was used for exchange and visualisation of the data representing the studied locations and pedestrian routes in the prototype.

In the begin of this research there was no clear view on modelling 3D pedestrian space therefore the second research question on modelling was defined:

2. How to model the 3D space to support access, interchange and egress walk-routing in multimodal travel advices?

As result of the literature study spaces and objects relevant for pedestrian routing were found in literature and described in paragraph 3.3.2. The accessible surface was used as basis for the new conceptual model as described in chapter 5. In this chapter the accessible surfaces and the topological relations between these surfaces are composed in a new conceptual model consisting of NodeSurfaces and LinkSurfaces described. Figure 14 show in a UML class view diagram what is needed to model a 3D geo-data model supporting access, interchange and egress walk route in multimodal travel advises. Modelling access areas, surfaces, logical routes, entrances, ramps, stairs lifts and elevators are crucial elements for the traveller. Also was established that pedestrian connecting directions for the accessible surfaces are important. The directions are important because directions are modelling the differences in going up or go down and are fundamental for the pedestrian access restrictions in the model. To determine up or down and the connection between the accessible surfaces 3D spatial functions were needed. 3D functionality was needed to distinguish an underpass from a quay above. In the model the differentiation in up/down not taking steepness of ramps and stairs into account was enough for the model. Although the model supports 3D calculation for walk durations only in the horizontal directions was sufficient for walk duration calculation. Defining accessibility to the various objects a traveller can experience in his pedestrian path in an accessibility matrix is a flexible way to support restrictions for travellers. At the start of the research how to model the different in mobility needs e.g. "unrestricted", "heavy luggage" and "wheelchair" on pedestrian routing was unclear. In the described new conceptual model only differences in parameters settings defined in cells of an accessibility matrix are needed to support pedestrian routing for different groups of travellers.

An objective of this research was to explore the capabilities of a DMBS (Oracle 11g) on storing pedestrian spaces and performing 3D routing in these spaces for individual travel advises. Therefore the third research question was defined at the start of this research:

3. Is a database supporting topological models like Oracle 11 suitable for supporting a geo-data model for access, interchange and egress walk routing?

The prototype results described in the previous chapters are showing that Oracle can support the proposed new pedestrian model. 3D storage of relevant pedestrian surfaces and connection pedestrian route are possible. The network and surfaces in Oracle where stored following the method as described in paragraph 6.1. The shortest path capabilities found in oracle and described in paragraph 6.2.1 are showing that these shortest path capabilities of Oracle 11 are sufficient for the pedestrian model. The functionalities to construct the geometric network from the 3D surfaces where limited. The limited support for 3D functions in Oracle was not expected at the start of this research. The height is ignored in most Oracle functions working with geodetic data. Distance calculation was not possible taking all the 3 dimensional directions into account on the geographic NodeSurfaces. As described in paragraph 6.1.1 using coordinate system with similar units of measures in all 3 directions to store NodeSurfaces would avoid this limitation.

8.2 Conclusions

In the conceptual design part of this research, it was possible to blend concepts found in literature and to compose a new conceptual model to support pedestrian routing in 3D taking travellers preferences in to account. The experimental part of the research using a combination of essential prototyping and (design) technical prototyping has showed that a working application can be build based on the proposed conceptual model. The experiments with the prototype validated the capabilities of the conceptual model to perform 3D pedestrian routing based on traveller preferences.

The intention of this research was to improve journey planners and therefore the two objectives where defined.

- *Design, develop and evaluate a geo-data model to support access, interchange and egress walk-routing in multimodal travel advices.*
- *Research the suitability of Oracle 11 to find walk routes in 3D geo-data relevant for the walk part of multimodal public transport journeys.*

It can be said that the first objective is almost reached, resulting in the pedestrian geo data model described in this thesis. In this research only limited evaluation of this model is done. The model is only tested with a small network of a medium side public transport place. Complex 3D situations and accessibility including escalators are not validated with the prototype. As described before evaluating this model with the data of more complex public transport place of national importance can prove the validity of this model. Implementing more public transport locations and combing it with existing street network data is can further validate this model. Also validating pedestrian route planning with the prototype with the focus on auxiliary activities (e.g. ticketing shopping) and routing inside vehicles could improve the geo data model. More research to validate the prototype with the intention to gain knowledge on the limitations of the described conceptual pedestrian model is needed to improve the conceptual model.

The second objective on the suitability of Oracle is answered by implementing a prototype. Unless the 3D limitations Oracle was suitable for the walk route finding. Important is that the prototype needed a lot of other tooling around Oracle to accomplish this. Tooling for 3D drawing, data conversion, data import and visualisation where needed to let Oracle do route finding. Other environments combining these functionalities might be more suitable for doing the job. Only implementing the conceptual model with in a different environment could give good inside in how suitable Oracle is compared to such other environments. From importance is that Oracle has some functionality related to this the subject where not touched by the prototype. Implementing constraints on network nodes and links in Java could improve the efficiency of optimal path calculation and as stated before the new functions to support large networks where not evaluated.

The achievement of this research is a novel 3D geo-data model of accessible pedestrian spaces in 3D to support pedestrian routing in multimodal public transport travel advices. Innovative in this geo data model is:

1. Identification, accessibility, visualisation and 3D spatial characteristics for pedestrian indoor and outdoor spaces are combined in one geo data model.

2. The concept of NodeSurfaces and LinkSurfaces to represent 3D pedestrian spaces in geo data models is novel. Using LinkSurfaces, having characteristics from links in a network and from OGC simple feature multi surfaces, is a simple, effective and well structured way to represent complex spaces used by pedestrian routes in a geo data model.

3. The flexible method of supporting accessibility and walk duration preferences for specific travellers to find optimal pedestrian routes using traveller's profiles, to enhance public travel advices.

Furthermore the model that supports the concept of walking freely on a surface "walk around" and a more strict way of walking on "hidden paths" over surfaces. In this way it combines the scene space and the network aspects of pedestrian routing. This research has also stressed that there is no standard route for an individual traveller with it individual needs. This in combination with the fact that adequate information helps to prepare the traveller for way finding, makes the importance of doing research and implementing this kind of geo-models is clear. As long as there are no multimodal public transport planners using a 3D geo-data model to support pedestrian routing this preparation is hardly possible. Therefore approaches to help individual travellers in the future getting this information are needed.

8.3 Outlook

The prototype approach followed in this research with a combination of essential prototyping and (design) technical prototyping as described in literature (Reeve, Petch 1999) has been useful. Using prototyping to gain practical knowledge on pedestrian routign has resulted in the described results in chapter 7. Further research to evaluate the feasibility and suitability of this new conceptual model for pedestrian routing with a prototype containing more data will increase this knowledge. No escalators and complex lift situations are modelled in this research. More practical prototyping with this kind of structures is needed to validate the suitability of the model for these situations. Implementing the conceptual model in an environment not restricted by the limitations found in Oracle is recommend to find other solutions for the limitations of the model connecting NodeSurfaces on edges and 3D distance calculations. Furthermore more inside in algorithms to define hidden path areas could be useful to limit labour when modelling complex public transport places. Visualisation of the results of the prototype could also be improved. Better visualisation of pedestrian routing is needed for practical use by travellers including showing underground objects in a clear way. Integrating sign information in the pedestrian model and visualisations is subject for further research.

Websites with travel information for public transportation must be able to generate travel advises fast and in volumes. The prototype did not focus performance and scalability aspects. The separation of logical steps performed in the prototype covering link assigning according to traveller needs, network loading and route calculation, as described in paragraph 6.2.1 followed by storing and visualisation of the optimal pedestrian route need to be reconsidered when developing a scalable and performing website application. Performance gain could be improved by keeping the pedestrian network in memory for all optimal route requests. A dedicated software program specific written for this functionality could optimize this process and outperform the Oracle network search capabilities used in the prototype.

During this research the 3D spatial data sets describing pedestrian locations were constructed based on aerial photos and existing map of these locations. This way of constructing 3D spatial datasets for pedestrian routing is time-consuming and prone to errors. The main purpose of these dataset was to illustrate and validate the proposed conceptual data model not to have up to date, accurate data about existing infrastructure. More efficient ways of capturing and collecting spatial information of 3D data to support pedestrian routing in public transport advices on a national scale must be found. Research could be done on utilities to support this capturing and exchange. Utilities to automatic create pedestrian surfaces out of existing building models could be beneficial in the creation of these datasets. Agreements on collection, exchange and maintenance of these 3D spatial data are important between the involved parties. A spatial data structure (SDI) supporting pedestrians routing around public transport is essential for this. The proposed conceptual model could be used as basis for a data exchange format used in such a SDI. Not only research on the data modelling aspects and organizing the maintenance of this data is needed. Research on the establishment of a spatial data infrastructure (SDI) for this would be a step towards a 3D geo-data model solution to support pedestrian routing in multimodal public transport travel advices on a national scale

REFERENCES

ARENS, C., STOTER, J. and VAN OOSTEROM, P., 2005. Modelling 3D spatial objects in a geo-DBMS using a 3D primitive. *Computers & Geosciences,* 31(2), 165-177.

BENNER, J., GEIGER, A. and LEINEMANN, K., 2005. Flexible generation of semantic 3D building models, *Proc of the 1st Intern. Workshop on Next Generation 3D City Models*.

BORNDÖRFER, R., GRÖTSCHEL, M. and PFETSCH, M.E., 2006. A Column-Generation Approach to Line Planning in Public Transport.

BRASEBIN, M., 2009. GeOxygene: An Open 3D Framework for the Development of Geographic Applications, 2009, 12th AGILE International Conference on Geographic Information Science 2009.

BUILDINGSMART, FAQ general questions, General questions about buildingSMART, IAI, and IFC [Homepage of buildingSMART], [Online]. Available: http://www.iai-tech.org/developers/faq/faq-general-questions [04/01, 2010].

CEN WG3 TC278, 2001-last update, CEN TC278, Reference Data Model For Public Transport, ENV12896 revised. Available: http://www.transmodel.org/.

CHOI, J. and LEE, J., 2008. 3D Geo-Network for Agent-based Building Evacuation Simulation.

GOOGLE, 2009a-last update, Touring in KML [Homepage of Google.com], [Online]. Available: http://code.google.com/intl/nl-NL/apis/kml/documentation/touring.html [04,06, 2010].

GOOGLE, 2009b-last update, Transit feed specification. Available: http://code.google.com/intl/nl/transit/spec/transit_feed_specification.html.

GOOGLE, 2009c-last update, Models [Homepage of Google.com], [Online]. Available: http://code.google.com/intl/nl-NL/apis/kml/documentation/models.html [04,06, 2010]

RÖGER, G., KOLBE, T.H., CZERWINSKI, A. and NAGEL, C., 2008. OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 1.0. 0, International OGC Standard. *Open Geospatial Consortium.*

GROTENHUIS, J.W., WIEGMANS, B.W. and RIETVELD, P., 2007. The desired quality of integrated multimodal travel information in public transport: Customer needs for time and effort savings. *Transport Policy,* 14(1), 27-38.

HOCHMAIR, H.H., 2008. Grouping of Optimized Pedestrian Routes for Multi-Modal Route Planning: A Comparison of Two Cities. *The European Information Society: Taking Geoinformation Science One Step Further,* 339.

HOCHMAIR, H., 2004. Towards a classification of route selection criteria for route planning tools. *Developments in Spatial Data Handling, Springer, Berlin,*, 481-492.

IFOPT, 2007. CEN TS287012008: Road traffic and transport telematics - Public transport - Identification of fixed objects in public transport.

KIKIRAS, P., TSETSOS, V., PAPATAXIARHIS, V., KATSIKAS, T. and HADJIEFTHYMIADES, S., 2009. User Modeling for Pedestrian Navigation Services. *Advances in Ubiquitous User Modelling,*, 111-133.

KNOWLES, N., 2008. *A Transmodel based XML schema for the Google Transit Feed Specification with a GTFS / Transmodel comparison.*

KOLBE, T.H., KÖNIG, G., NAGEL, C. and STADLER, A., 2009. 3D GeoDatabase for CityGML Version 2.0.1. Berlin: Institute for Geodesy and Geoinformation Science Technische Universität Berlin.

LEE, J., 2004. A spatial access-oriented implementation of a 3-D GIS topological data model for urban entities. *GeoInformatica,* 8(3), 237-264.

LEE, J. and ZLATANOVA, S., 2008. A 3D data model and topological analyses for emergency response in urban areas. *Geospatial Information Technology for Emergency Response,* Geospatial information technology for emergency response (ISPRS book, 143. series) Taylor & Francis Group, London, UK, pp143-168.

LEE, Y.C., 2005. *Pedestrian Walking and Choice Behavior on Stairways and Escalators in Public Transport Facilities*, Department of Transport & Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology.

MEIJERS, M., ZLATANOVA, S. and PFEIFER, N., 2005. 3D geoinformation indoors: structuring for evacuation, *Proceedings of Next generation 3D city models,* 2005, pp21-22.

MURRAY, C., AN, N., GONG, H., RAVADA, S. and WANG, J., 2007. Oracle® Spatial Topology and Network Data Models Developer's Guide 11*g* Release 1 (11.1) B28399-02. Oracle Corporation.

MURRAY, C., AN, N., GONG, H., RAVADA, S. and WANG, J., 2006. Oracle® Spatial Topology and Network Data Models 10*g* Release 2 (10.2) Part Number B14256-02. Oracle Corporation.

MURRAY, C., 2010. Oracle® SpatialDeveloper's Guide11g Release 2. March 2010 edn. Redwood City: Oracle Corporation.

OGC, 2008. OGC® KML. Reference number of this OGC® project document: OGC 07-147r2 Version: 2.2.0 edn. Open Geospatial Consortium Inc.

OGC, 1999. OpenGIS Simple Features Specification For SQL. OpenGIS Project Document 99-049 edn. Open GIS Consortium, Inc.

REEVE, D. and PETCH, J., 1999. GIS, Organisations and People: A Socio-technical Approach. CRC Press.

RÜETSCHI, U.J., 2007. *Wayfinding in Scene Space Modelling Transfers in Public Transport*, University Zurich.

RÜETSCHI, U.J. and TIMPF, S., 2005. Modelling wayfinding in public transport: network space and scene space. *Spatial Cognition IV,*, pp24-41.

SLINGSBY, A. and RAPER, J., 2008. Navigable Space in 3D City Models for Pedestrians. *Advances in 3D Geoinformation Systems,*, pp49-64.

STEVENS, M. and CHOI, J., 2006. CAD Data Conversion to a Node-Relation Structure for 3D Sub-Unit Topological Representation. *Journal of the Korean Geographical Society,* 41(2), pp188-194.

WANG, J.C., 2009. *A Load-On-Demand Approach to Handling Large Networks in the Oracle Spatial Network Data Model.* Redwood Shores, USA: Oracle Corporation.

APPENDICES

A. Applications used for prototype

The following applications were used implementing the prototype in chapter 6.

Application	Version	URL from supplier
Google SketchUp	6.4.247	SketchUp.google.com
Oracle Database	11g 11.2.0.1.0 - 64bit	www.oracle.com
Enterprise architect	7.5.800	www.sparxsystems.com
FME	2010 build 6201	www.safe.com
SQL developer	1.5.5	www.oracle.com
Google earth	5.1.3534.0411	earth.google.com
Microsoft Visual C# 2008	9.30729 sp1	Microsoft.com
GraphSharp	V 1.0	graphsharp.codeplex.com

B. IFOPT component stop place hierarchy

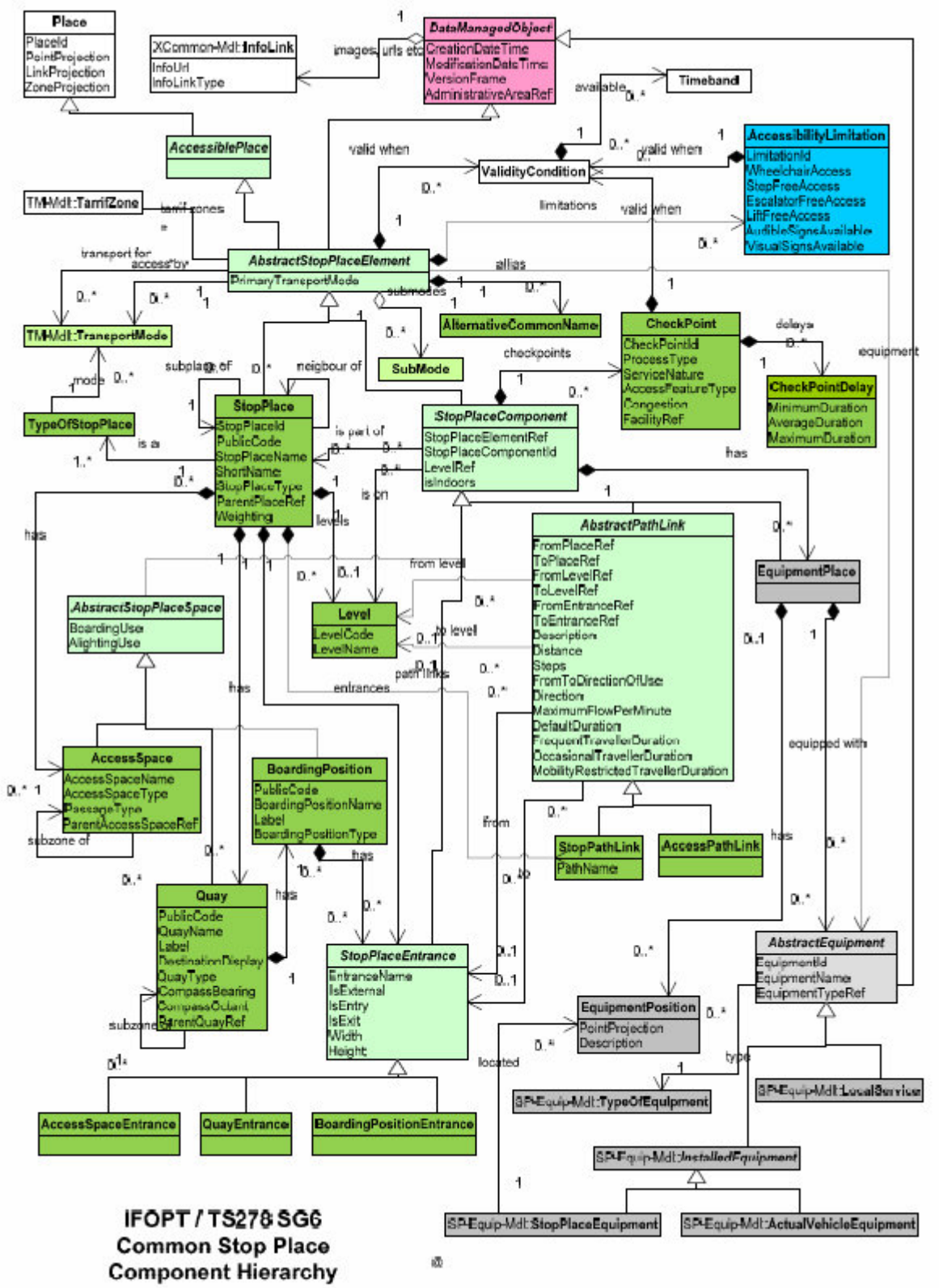


Figure 46 component stop place hierarchy (IFOPT 2007)

The Stop Place model is part of the IFOPT standard described in paragraph 3.3.2

C. CityGML surface implementation in a spatial database (Kolbe, König et al. 2009)

Paragraph 3.5.1 describes CityGML. In CityGML Berlin's database, the modelling of two-dimensional and three-dimensional geometry types is handled in a simplified way. All surface-based geometries are stored as polygons, which are aggregated to MultiSurfaces, CompositeSurfaces, TriangulatedSurfaces, Solids, MultiSolids, as well as CompositeSolids accordingly. This simplification substitutes the more complex representation used for those GML geometry classes in grey blocks in UML diagram below. Mapping the UML diagram to the relational schema now requires only one table (SURFACE_GEOMETRY)

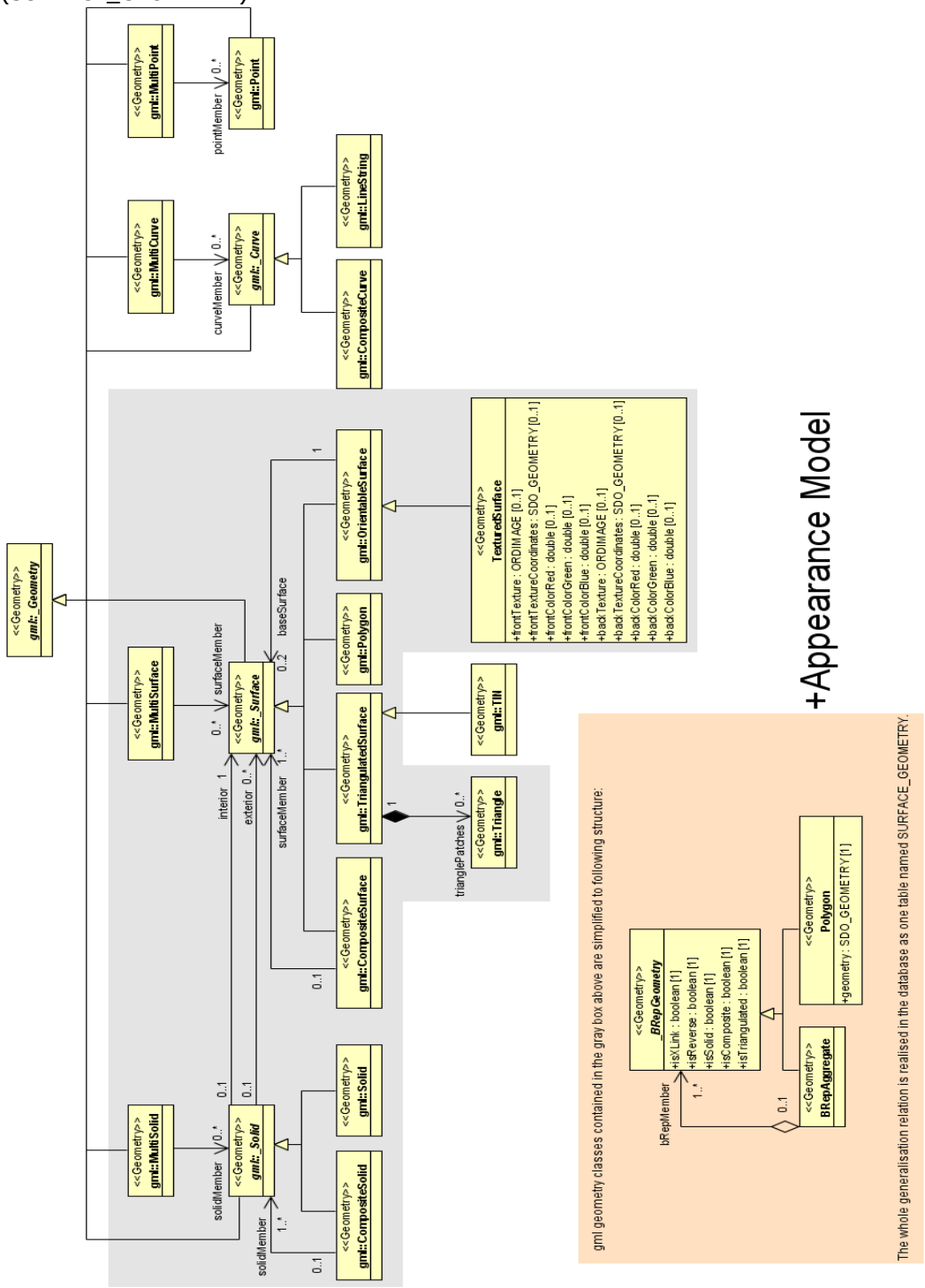


Figure 47 UMLCityGML surface implementation in a spatial database (Kolbe, König et al. 2009)

D. UML diagram shown overview of IFC structure

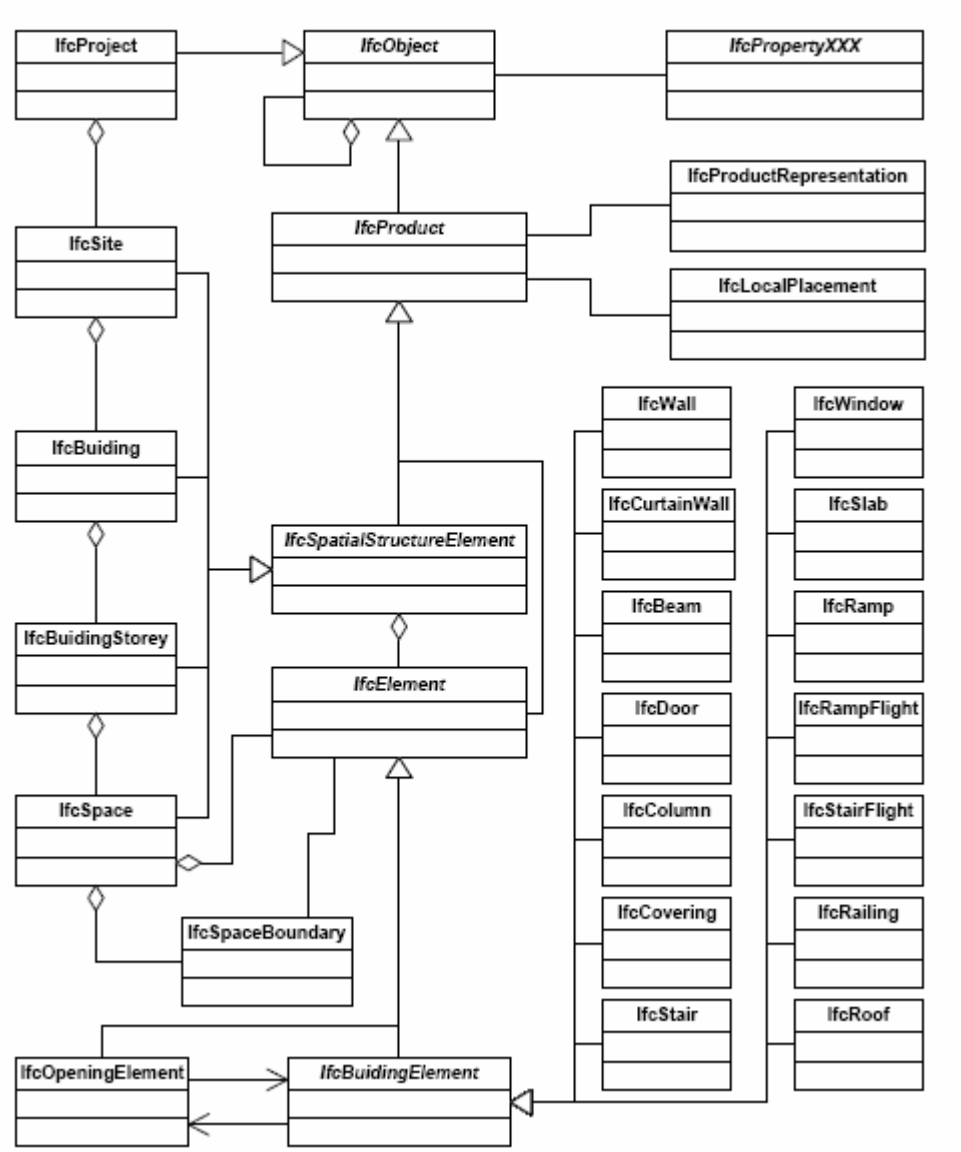


Figure 48 UML diagram shown overview of IFC structure (Benner, Geiger et al. 2005)**.**

Industry Foundation Classes a standard for building models is described in paragraph 3.5.2.

E. Aspects influencing a optimal pedestrian route

In literature (Lee 2005) is described that route choice is based on travel time and time pressure, physical effort safety and conform personnel characteristics and vicinity are seen as important criteria. To understand the complexity of finding an optimal route for an individual traveller in a certain situation these aspects are described in detail. From these aspects accessibility and walk duration are selected as important in this model (see paragraph 4.1.)

Travel time and time pressure

Travellers always value the time spent on a trip. Thus pedestrians prefer the route with the least travel time. To which extent the travel time is perceived by pedestrians is mainly determined by the trip purpose or the time pressure facing by pedestrians. Normally, the travel time is more highly valued by commuters than tourists. However, taking into account human limitations, the perception towards the travel time may depend on the scale of time as well. Than mean the value of time of a 10-minute walk may be different from that of a 1-hour walk. Although pedestrians may increase their speeds to shorten their travel time, this requires extra human energy. Thus the choice behaviour is influence by the relative delay as well as the total travel time of alternative routes.

Physical efforts

Walking consumes energy. Thus pedestrians prefer the route involving least physical effort. With respect to movement in the vertical dimension, the physical effort involved is determined by factors such as the configuration of infrastructure (slope and length), directions of movement (upwards or downwards), and the presence of luggage. Escalators and lifts provide chances for pedestrians to change levels without consuming energy, which is highly valued by pedestrians with poor physical fitness or with luggage.

Safety and comfort

Walking involves not only the movement of extremes but mental activities for interacting with others and the environment. External influences such as unfavourable terrains (e.g. stairs) and the presence of conflicts (e.g. opposing traffic) increase mental efforts required and consequently decrease travel comfort. Escalators unify the movement direction of flows, and, consequently enhance travel safety and comfort. Finally, conflicting traffic on the approaching way leading to the facilities will decrease travel comfort as well.

Personal characteristics

The perception towards physical effort as well as safety and comfort may depend on personal characteristics such as gender, age or physical fitness, habit, and health concerns.

Vicinity

It is observed that pedestrians have higher inclination towards the nearest facilities. Thus based on their approaching direction, they may choose the first facility encountered on the way.

F. Travel advice from GIMA locations Wageningen to Utrecht

The following travel advice was used as journey between the Forum building in Wageningen and Utrecht University library, via bus stop "Droevendaalse steeg (direction Ede)", train station Ede-Wageningen and Utrecht Central station. (Chapter 4)

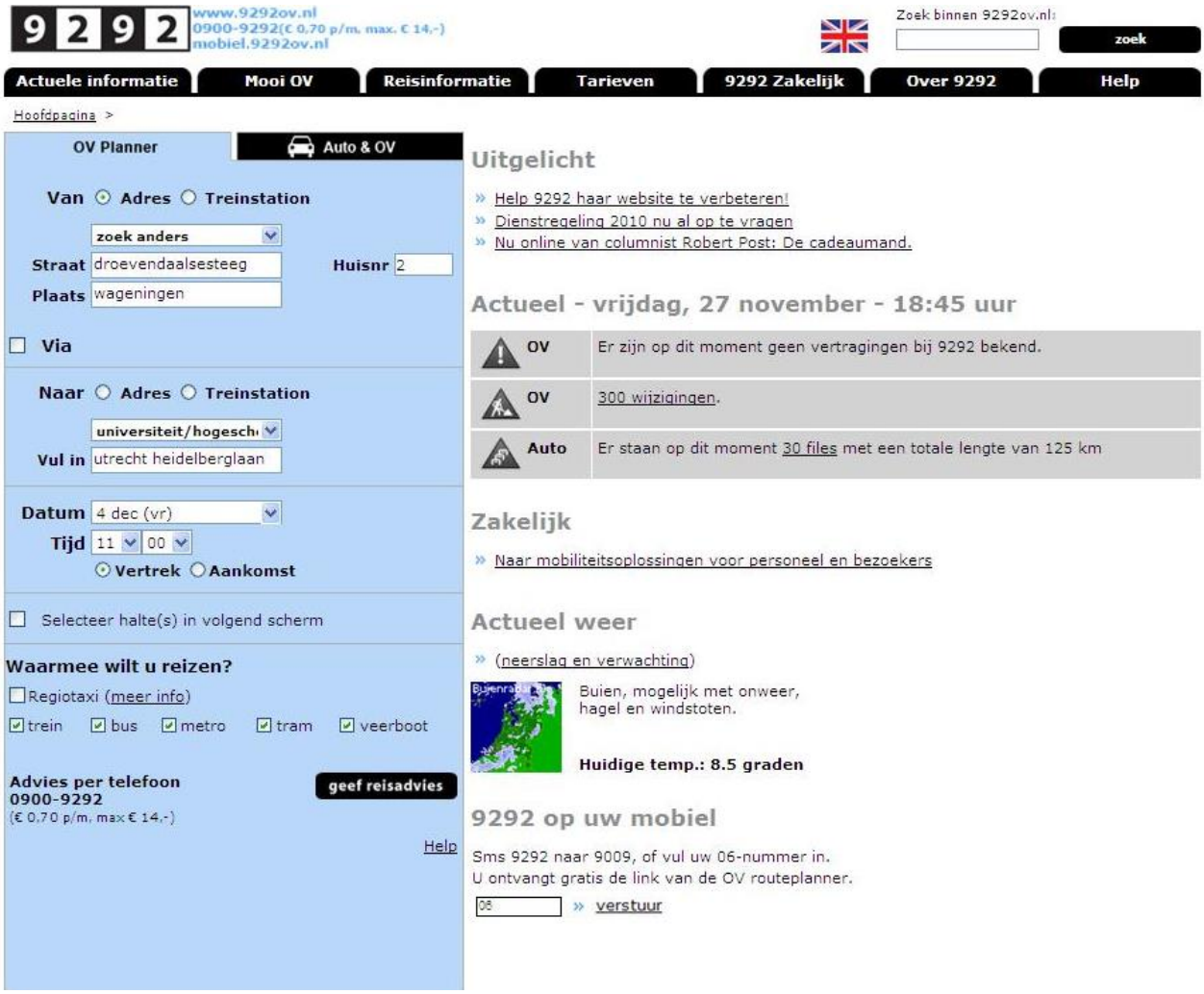


Figure 49 Entry webpage from www.9229ov.nl

OV advies

terugreis | wijzig reis | nieuw advies

Van droevendaalsesteeg 2, wageningen
Naar universiteit/hogeschool utrecht heidelberglaan
Datum vrijdag, 4 december 2009 11:00

Vertrek	9:01	9:11	9:30	9:45	10:00	« Eerder
Aankomst	10:07	10:34	10:37	11:04	11:07	» Later
Reistijd	1:06	1:23	1:07	1:19	1:07	« Eerste reismogelijkheid
Overstappen	2	2	2	2	2	» Laatste reismogelijkheid

Vertrek	Van	Naar	Aankomst	Hoe
9:45	droevendaalsesteeg wageningen	Droevendaalsesteeg Wageningen	9:51	lopen 6 min
9:51	Droevendaalsesteeg Wageningen	Station Ede Wageningen Zuid	10:04	bus 88 Veolia richting Ede
10:10	treinstation Ede-Wageningen spoor 3	treinstation Centraal Utrecht spoor 5	10:37	IC intercity NS richting Den Helder
10:37	treinstation Centraal Utrecht	Busstation Stadsvervoer Utrecht	10:47	lopen 10 min
10:48	Busstation Stadsvervoer Utrecht	Heidelberglaan Utrecht	11:04	stadsbus 12 GVU richting Uithof UMC AZU

Figure 50 Travel advice from www.9292ov.nl

G. Insert statement using SDO_UTIL.FROM_KMLGEOMETRY

Paragraph 6.1.1 The KML files with the NodeSurfaces where parsed with a C# program and converted to SQL statements to insert the data in the table NodeSurface. To convert the data from KML to Oracle surfaces the Oracle utility function "FROM_KMLGEOMETRY" is used:

```
INSERT INTO NODESURFACE (NODE_ID, ACTIVE,NODE_NAME, GEOMETRY3D)
VALUES(1,'Y','fietspadzebraPON', SDO_UTIL.FROM_KMLGEOMETRY
('<Polygon><extrude>1</extrude><tessellate>1</tessellate><altitudeMode>relativeToGround</altitu
deMode><outerBoundaryIs><LinearRing><coordinates>
5.668831595491851,51.98792812451821,2 5.668865823964326,51.98788596623955,2
5.668928241374767,51.98789464551311,2 5.668980591523003,51.98790704475545,2
5.669032941699257,51.98792068395456,2 5.669077238107658,51.9879343230804,2
5.669099386366625,51.98794548253271,2 5.669155763674707,51.98796408140326,2
5.669204087106903,51.9879789604316,2 5.669242343128593,51.9879888797248,2
5.669224222247817,51.98800623906341,2 5.669212141698557,51.98801863863139,2
5.669200061101494,51.9880297982529,2 5.669179308002914,51.98805290669646,2
5.669160118013816,51.98805228079042,2 5.66914876832209,51.98804279521309,2
5.669150389634375,51.98803380884515,2 5.668831595491851,51.98792812451821,2
</coordinates></LinearRing></outerBoundaryIs></Polygon>'));
```

H. Script to create NodeSurfaces and LinkSurfaces network in Prototype

Paragraph 6.1.1. desribes the creation of the network. This psql script was used:

```
-----------------------------------------------------------------
-- CREATE network tables --
-----------------------------------------------------------------
execute SDO_NET_MEM.NETWORK_MANAGER.DROP_NETWORK('PEDMODEL');

execute SDO_NET.DROP_NETWORK('PEDMODEL');
            execute        SDO_NET.CREATE_SDO_NETWORK('PEDMODEL',        1,                TRUE,
'NODESURFACE','networkGEOMETRY'    ,    'distance','LinkSurface','networkGEOMETRY',    'delay',
'route','networkGEOMETRY', 'routeElement');
            SELECT SDO_NET.VALIDATE_NODE_SCHEMA('PEDMODEL') FROM DUAL;


alter table NOdesurface
ADD
GEOMETRY3D  MDSYS.SDO_GEOMETRY ;

alter table NOdesurface
ADD
Z  Number ;

alter table linksurface
ADD
ZDirection  VARCHAR2(1) ;

alter table NOdesurface
ADD
nodeSurfaceID  Integer NULL;

CREATE UNIQUE INDEX UNIQUENODESURFACE ON NODESURFACE ('NODESURFACEID') ;


alter table linksurface
add CONSTRAINT "FK_ACCESTYPE" FOREIGN KEY ("LINK_LEVEL")
         REFERENCES "AccesType" ("accesTypeID") ENABLE

--alter table ROUTE
--ADD
--GEOMETRY3D  MDSYS.SDO_GEOMETRY;



            INSERT  INTO  NODESURFACE  (NODE_ID,  ACTIVE,    NODE_NAME,  GEOMETRY3D)    VALUES
(1,'Y','fietspadzebraPON',                                          SDO_UTIL.FROM_KMLGEOMETRY
('<Polygon><extrude>1</extrude><tessellate>1</tessellate><altitudeMode>relativeToGround</altitu
deMode><outerBoundaryIs><LinearRing><coordinates>
                 5.668831595491851,51.98792812451821,2   5.668865823964326,51.98788596623955,2
5.668928241374767,51.98789464551311,2

insert statement removed for documentation only

INSERT INTO NODESURFACE (NODE_ID, ACTIVE,   NODE_NAME, GEOMETRY3D)   VALUES    (29,'Y','weg2',
SDO_UTIL.FROM_KMLGEOMETRY
('<Polygon><extrude>1</extrude><tessellate>1</tessellate><altitudeMode>relativeToGround</altitu
deMode><outerBoundaryIs><LinearRing><coordinates>
                 5.664365990373863,51.98576689632159,2   5.664438526620693,51.98578810856108,2
5.664208865344835,51.98611482121979,2                    5.664129232792003,51.98610024275786,2
5.664365990373863,51.98576689632159,2
                </coordinates></LinearRing></outerBoundaryIs></Polygon>'));

update nodesurface
set node_type = '#AccessSpace';


delete from USER_SDO_GEOM_METADATA where table_name = 'LINKSURFACE';
delete from USER_SDO_GEOM_METADATA where table_name = 'NODESURFACE';
delete from USER_SDO_GEOM_METADATA where table_name = 'ROUTE';
```

```
INSERT INTO user_sdo_geom_metadata
    (TABLE_NAME,
     COLUMN_NAME,
     DIMINFO,
     SRID)
  VALUES (
  'NODESURFACE',
  'GEOMETRY3D',
  SDO_DIM_ARRAY(
   MDSYS.SDO_DIM_ELEMENT('X', 5.000000000000000, 6.0000000000000000, 0.000000000000005),
                              MDSYS.SDO_DIM_ELEMENT('Y',                    51.00000000000000,
53.000000000000000,0.000000000000005),
                              MDSYS.SDO_DIM_ELEMENT('Z',0.00000000000000    ,10.00000000000000,
0.1))  , 4326

);

INSERT INTO user_sdo_geom_metadata
    (TABLE_NAME,
     COLUMN_NAME,
     DIMINFO,
     SRID)
  VALUES (
  'ROUTE',
  'GEOMETRY3D',
  SDO_DIM_ARRAY(
  MDSYS.SDO_DIM_ELEMENT('X', 5.000000000000000, 6.0000000000000000, 0.000000000000005),
                              MDSYS.SDO_DIM_ELEMENT('Y',                    51.00000000000000,
53.000000000000000,0.000000000000005),
                              MDSYS.SDO_DIM_ELEMENT('Z',0.00000000000000    ,10.00000000000000,
0.1))

                              , 4326

);

update NODESURFACE c
set C.GEOMETRY3D.SDO_SRID=4326;


-------------------------------------------------------------------
-- CREATE THE SPATIAL INDEX --
-------------------------------------------------------------------
CREATE INDEX NODESURFACE_spatial_idx
 ON NODESURFACE(GEOMETRY3D)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=2');

CREATE INDEX ROUTE_spatial_idx
 ON ROUTE(GEOMETRY3D)
 INDEXTYPE IS MDSYS.SPATIAL_INDEX
 PARAMETERS ('sdo_indx_dims=2');


-------------------------------------------------------------------
-- Fill ede wageingen stopplace   --
-------------------------------------------------------------------

INSERT INTO   nodesurface  (  node_id,  node_name, node_type,  active, GEOMETRY3D)
select  rownum  +  29,   rtrim(layer)  ||   '  '  ||  rtrim(partno)  ,  rtrim(layer),  'Y',GEOM  from
polygons  ;

Update NODESURFACE c
set Z= (select round ( (min(t.Z) + max (t.Z))/2 , 0)
   FROM NODESURFACE  ,
   TABLE(SDO_UTIL.GETVERTICES(c.GEOMETRY3D)) t
  where node_id = c.node_id ) ;

-------------------------------------------------------------------
-- Fill linksurface   --
-------------------------------------------------------------------
INSERT  INTO     linksurface   (  link_id,   link_name,  start_node_id  ,  end_node_id,  active  ,
Zdirection)
select rownum,   a.node_name || ' -> ' || b.node_name || ' ' ||    CASE
    WHEN  a.Z < b.Z  THEN '(Up)'
    WHEN  a.Z > b.Z  THEN '(Down)'
```

```
    ELSE ''
     END CASE
      ,
      a.node_id, b.node_id,'Y' ,
        CASE
    WHEN  a.Z < b.Z  THEN 'U'
    WHEN  a.Z > b.Z THEN 'D'
    ELSE 'L'
     END CASE
        from NODESURFACE a,  NODESURFACE b
where SDO_GEOM.RELATE(
    a.GEOMETRY3D,
     'ANYINTERACT' ,
     b.GEOMETRY3D,
   0.000000000000005
     )= 'TRUE' AND a.node_id <> b.node_id;

update linksurface T1
set link_level =( select T3.accestypeid
from nodesurface T2
join ACCESTYPE_NODETYPE T3 on t3.node_type = T2.node_type
where (    T2.node_id = T1.end_node_id and T1.zdirection= T3.zdirection) )
   ;

-------------------------------------------------------------------
-- empty network cost node  --
-------------------------------------------------------------------
update NODESURFACE c
set distance =  0 ;

-------------------------------------------------------------------
-- empty network cost path  --
-------------------------------------------------------------------
update Linksurface c
set delay = 0 ;

-------------------------------------------------------------------
-- check network  --
-------------------------------------------------------------------
SELECT                                                  node_id,                        node_name,
c.GEOMETRY3D.Get_GType(),SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT(c.GEOMETRY3D                 ,
SDO_DIM_ARRAY(
  MDSYS.SDO_DIM_ELEMENT('X', 5.000000000000000, 6.0000000000000000, 0.5),
                              MDSYS.SDO_DIM_ELEMENT('Y',                     51.00000000000000,
53.000000000000000,0.5),
                              MDSYS.SDO_DIM_ELEMENT('Z',0.00000000000000    ,10.00000000000000,
0.5))

                              , 4326)
   FROM nodesurface c;

select * from ALL_SDO_NETWORK_METADATA;
select * from ALL_SDO_NETWORK_CONSTRAINTS;

SELECT SDO_NET.VALIDATE_NETWORK('PEDMODEL') FROM DUAL;
SELECT SDO_NET.GET_NO_OF_LINKS('PEDMODEL') FROM DUAL;
SELECT SDO_NET.GET_NO_OF_NODES('PEDMODEL') FROM DUAL;
```

I. Procedure Makeroute

Paragraph 6.2.1: A stored procedure MAKEROUTE is developed during this research to calculate the routes according to the travellers needs and speed profile. This procedure performs the steps 3, 4, 5 and 6 as described in the conceptual model in paragraph 5.8. A full listing of this PL/SQL MAKEROUTE procedure is

```
create or replace
PROCEDURE Makeroute (pathname  IN VARCHAR2,
                                     fromnode  IN NUMBER,
                                     tonode    IN NUMBER,
                                     needtype  IN NUMBER,
                                     speedtype IN NUMBER)
IS
  net_mem  VARCHAR2(100);
  res_num  NUMBER;
  rest     VARCHAR2(1000);
  res_geom SDO_GEOMETRY;
  res_val  VARCHAR2(1000);
  cost     NUMBER;
--path_id NUMBER;
--res_array SDO_NUMBER_ARRAY;
--indx NUMBER;
BEGIN
  UPDATE linksurface
  SET    active = 'N',
         delay = 0;

  UPDATE linksurface t1
  SET    active = 'Y',
         delay = (SELECT "Timeduration"
                  FROM   "LINKSpeedVIEW"
                  WHERE  "speedTypeID" = speedtype
                         AND "LINK_ID" = t1.link_id)
  WHERE  link_level IN (SELECT "accesTypeID"
                        FROM   "AccesMatrix"
                        WHERE  "mobilityNeedID" = needtype);

  net_mem := 'PEDMODEL';

  --Read in the network.
  sdo_net_mem.network_manager.Read_network('PEDMODEL', 'TRUE');

  rest := sdo_net_mem.network_manager.list_networks;

  dbms_output.Put_line('The current in-memory network(s) is/are: '
                       || rest);

  res_num := sdo_net_mem.network_manager.Shortest_path('PEDMODEL', fromnode, tonode, NULL);

  dbms_output.Put_line('The shortest path from startnode to endnode is path ID: '
                       || res_num);

  -- Get some characteristics of the resultant path.
  IF res_num IS NULL THEN
    dbms_output.Put_line('No Path found');

    sdo_net_mem.network_manager.Drop_network('PEDMODEL');

    INSERT INTO noroute
                (path_name,
                 path_type,
                 start_node_id,
                 end_node_id)
    VALUES      ( pathname,
                 speedtype,
                 fromnode,
                 tonode);

    RETURN;
  END IF;

  dbms_output.Put_line('The characteristics of path '
```

```
                      || res_num);

  cost := sdo_net_mem.path.Get_cost('PEDMODEL', res_num);

  dbms_output.Put_line('The path '
                      || res_num
                      || ' has cost: '
                      || cost);

  res_val := sdo_net_mem.path.Is_closed('PEDMODEL', res_num);

  dbms_output.Put_line('Is path  || res_num ||  closed ? '
                      ||res_val);

  -- Add the resultant path to the network.
  sdo_net_mem.path.Set_name('PEDMODEL', res_num, pathname);

  sdo_net_mem.path.Set_type('PEDMODEL', res_num, speedtype);

  sdo_net_mem.network.Add_path('PEDMODEL', res_num);

  sdo_net_mem.network_manager.Write_network('PEDMODEL');

  sdo_net_mem.network_manager.Drop_network('PEDMODEL');

  SELECT Substr(MAX(Replace(Sys_connect_by_path(link_id, '/'), '/', ' ')), 2)
  INTO   rest
  FROM   (SELECT path_id,
                 seq_no,
                 link_id,
                 Row_number() over (PARTITION BY path_id ORDER BY seq_no ) row#
          FROM   routeelement
          WHERE  path_id = res_num)
  START WITH row# = 1
  CONNECT BY PRIOR path_id = path_id
                   AND PRIOR row# = row# - 1
  GROUP  BY path_id;

  dbms_output.Put_line('Path: '
                      || rest);
END;
```

J. SQL script to test Makeroute

In Paragraph 6.2.1 the following procedure is mentioned:

```
DECLARE

fromnode NUMBER;
tonode NUMBER;
fromnodename VARCHAR2(200);
tonodename VARCHAR2(200);
cursor get_speedtype   is  SELECT "Name", "speedTypeID" FROM "SpeedType"
-- where "speedTypeID" =3
order by"speedTypeID" ;
cursor get_needs       is  SELECT "Name", "mobilityNeedID" FROM "MobilityNeed"
--where "mobilityNeedID"= 0
--where "mobilityNeedID"= 6
where "mobilityNeedID"= 1
order by "mobilityNeedID";
speedname "SpeedType"."Name"%TYPE;
speedid   "SpeedType"."speedTypeID"%TYPE;
needid NUMBER;
needname varchar2(200);

BEGIN
DBMS_OUTPUT.ENABLE(1000000);

-- bus to train
 fromnodename := '#BoardingPosition_positionAtBusStop 0';
 tonodename := '#BoardingPosition_positionOnRailPlatform 0';

-- train to bus
-- tonodename := '#BoardingPosition_positionAtBusStop 0';
-- fromnodename := '#BoardingPosition_positionOnRailPlatform 0';

--  other locations
--'#CheckPoint_lift 0'; -- '#AccessSpace_Underpass 0'; --;
--'#AccessSpace_Underpass 0'; --
-- '#AccessSpace 9'; --'#CheckPoint_ramp 0';
--'#CheckPoint_lift 0' ;
-- '#BoardingPosition_positionOnRailPlatform 0' ;

--sdo_net_mem.network_manager.drop_network('PEDMODEL');
delete from routeelement;
delete from route;
delete from noroute;

select NODE_ID INTO  fromnode from nodesurface where  node_name =  fromnodename ;
select NODE_ID INTO  tonode from nodesurface where node_name  = tonodename;

      -- Open first cursor
      open get_speedtype ;
      loop
          fetch get_speedtype  into speedname, speedid;
          EXIT WHEN get_speedtype %NOTFOUND;
          open get_needs ;
          loop
            fetch get_needs into needname, needid;
              EXIT WHEN get_needs %NOTFOUND;
                DBMS_OUTPUT.PUT_LINE ('test'|| ' '|| needname ||' '|| speedname );
                MAKEROUTE(  fromnodename || ' to '|| tonodename|| needid || '-'|| speedid || '
'|| needname ||' '|| speedname  , fromnode,tonode, needid,speedid );

         end loop;
         close get_needs;
       end loop;
       close get_speedtype;
 END;
```

K. Tables for network and surface

In Paragraph 6.1.1 the following tables are mentioned:

```
  CREATE TABLE "NODESURFACE"
   (  "NODE_ID" NUMBER,
      "NODE_NAME" VARCHAR2(200 BYTE),
      "NODE_TYPE" VARCHAR2(200 BYTE),
      "ACTIVE" VARCHAR2(1 BYTE),
      "PARTITION_ID" NUMBER,
      "NETWORKGEOMETRY" "MDSYS"."SDO_GEOMETRY" ,
      "DISTANCE" NUMBER,
      "GEOMETRY3D" "MDSYS"."SDO_GEOMETRY" ,
      "Z" NUMBER,
      "NODESURFACEID" NUMBER(*,0),
       PRIMARY KEY ("NODE_ID")

CREATE TABLE LINKSURFACE"
   (  "LINK_ID" NUMBER,
      "LINK_NAME" VARCHAR2(200 BYTE),
      "START_NODE_ID" NUMBER NOT NULL ENABLE,
      "END_NODE_ID" NUMBER NOT NULL ENABLE,
      "LINK_TYPE" VARCHAR2(200 BYTE),
      "ACTIVE" VARCHAR2(1 BYTE),
      "LINK_LEVEL" NUMBER,
      "NETWORKGEOMETRY" "MDSYS"."SDO_GEOMETRY" ,
      "DELAY" NUMBER,
      "BIDIRECTED" VARCHAR2(1 BYTE),
      "ZDIRECTION" VARCHAR2(1 BYTE),
       PRIMARY KEY ("LINK_ID")
CONSTRAINT "FK_ACCESTYPE" FOREIGN KEY ("LINK_LEVEL")
        REFERENCES "AccesType" ("accesTypeID") ENABLE
```

L. Tables for traveller profile mobility needs and speedtype

In Paragraph 6.1.2 the following tables are mentioned:

```
CREATE TABLE "Traveller Profile"
   (  "traveller ProfileID" NUMBER(*,0) NOT NULL ENABLE,
      "mobilityNeedID" NUMBER(*,0) NOT NULL ENABLE,
      "speedTypeID" NUMBER(*,0) NOT NULL ENABLE,
       CONSTRAINT "PK_T Profile" PRIMARY KEY ("traveller ProfileID")
       CONSTRAINT     "FK_Traveller     P_MobilityNeed"     FOREIGN     KEY
("mobilityNeedID")
        REFERENCES "MobilityNeed" ("mobilityNeedID") ENABLE,
       CONSTRAINT    "FK_Traveller    Profile_SpeedType"    FOREIGN    KEY
("speedTypeID")
        REFERENCES "SpeedType" ("speedTypeID")

  CREATE TABLE "MobilityNeed"
   (  "Name" VARCHAR2(200 BYTE),
      "mobilityNeedID" NUMBER(*,0) NOT NULL ENABLE,
       CONSTRAINT "PK_MobilityNeed" PRIMARY KEY ("mobilityNeedID")


CREATE TABLE "SpeedType"
   (  "Name" VARCHAR2(200 BYTE),
      "speedTypeID" NUMBER(*,0) NOT NULL ENABLE,
       CONSTRAINT "PK_SpeedType" PRIMARY KEY ("speedTypeID")
```

```
CREATE TABLE "AccesMatrix"
   (  "accesMatrixID" NUMBER(*,0) NOT NULL ENABLE,
      "accesTypeID" NUMBER(*,0) NOT NULL ENABLE,
      "mobilityNeedID" NUMBER(*,0) NOT NULL ENABLE,
       CONSTRAINT "PK_AccesMatrix" PRIMARY KEY ("accesMatrixID")
       CONSTRAINT "FK_AccesMatrix_AccesType" FOREIGN KEY ("accesTypeID")
        REFERENCES "AccesType" ("accesTypeID") ENABLE,
       CONSTRAINT       "FK_AccesMatrix_MobilityNeed"       FOREIGN       KEY
("mobilityNeedID")
        REFERENCES ""MobilityNeed" ("mobilityNeedID") ENABLE
   )


  CREATE TABLE "SpeedMatrix"
   (  "Speed" NUMBER(8,2),
      "CalcMethode" NUMBER(*,0),
      "accesTypeID" NUMBER(*,0) NOT NULL ENABLE,
      "speedMatrixID" NUMBER(*,0) NOT NULL ENABLE,
      "speedTypeID" NUMBER(*,0) NOT NULL ENABLE,
       CONSTRAINT "PK_SpeedMatrix" PRIMARY KEY ("speedMatrixID")
  )
       CONSTRAINT "FK_SpeedMatrix_AccesType" FOREIGN KEY ("accesTypeID")
        REFERENCES "AccesType" ("accesTypeID") ENABLE,
       CONSTRAINT "FK_SpeedMatrix_SpeedType" FOREIGN KEY ("speedTypeID")
        REFERENCES "SpeedType" ("speedTypeID") ENABLE
   )

CREATE TABLE "AccesType"
   (  "UP" VARCHAR2(1 BYTE),
      "Name" VARCHAR2(200 BYTE),
      "accesTypeID" NUMBER(*,0) NOT NULL ENABLE,
       CONSTRAINT "PK_AccesType" PRIMARY KEY ("accesTypeID")

  CREATE TABLE "ACCESTYPE_NODETYPE"
   (  "NODE_TYPE" VARCHAR2(200 BYTE),
      "ZDIRECTION" VARCHAR2(1 BYTE),
      "ACCESTYPEID" NUMBER
   ) SEGMENT CREATION IMMEDIATE
```

M. Tables for storing routing results

In paragraph 6.2.2 the tables are describe for storing results:

```
CREATE TABLE "ROUTE"
   (  "PATH_ID" NUMBER,
      "PATH_NAME" VARCHAR2(200 BYTE),
      "PATH_TYPE" VARCHAR2(200 BYTE),
      "START_NODE_ID" NUMBER NOT NULL ENABLE,
      "END_NODE_ID" NUMBER NOT NULL ENABLE,
      "COST" NUMBER,
      "SIMPLE" VARCHAR2(1 BYTE),
      "GEOMETRY3D" "MDSYS"."SDO_GEOMETRY" ,
       PRIMARY KEY ("PATH_ID")

CREATE TABLE "ROUTEELEMENT"
   (  "PATH_ID" NUMBER NOT NULL ENABLE,
      "LINK_ID" NUMBER NOT NULL ENABLE,
      "SEQ_NO" NUMBER,
```

```
PRIMARY KEY ("PATH_ID", "LINK_ID", "SEQ_NO")
```

N. Table created with FME for importing surface geometries with SketchUp layer names

Paragraph 6.1.1: A conversion is FME (Deaggregate) is done to create surfaces out the multiGeometry objects in the kmz files. The Google SketchUp layer name is used in combination with the sequence number of the surface in the layer (part number). The definitions of the table, named polygons, use to temporary store these surfaces is:

```
CREATE TABLE "POLYGONS"
   (  "LAYER" CHAR(80 BYTE),
      "PARTNO" CHAR(30 BYTE),
      "GEOM" "MDSYS"."SDO_GEOMETRY"
   )
```

O. Views used in prototype

The following views are use in the prototype (paragraph 6.2.3)

1) LINKSPEEDVIEW

```
  CREATE  OR  REPLACE  FORCE  VIEW  "LINKSpeedVIEW"  ("LINK_ID",  "LINK_LEVEL",  "LINK_NAME",
"START_NODE_ID",  "NODE_NAME",  "NODE_TYPE",  "END_NODE_ID",  "END_NODE_NAME",  "END_NODE_TYPE",
"Calcmethode", "Speed", "speedTypeID", "Distance", "Timeduration", "GEOMETRY3D") AS
  SELECT t1.link_id,
         t1.link_level,
         t1.link_name,
         t1.start_node_id,
         t2.node_name,
         t2.node_type,
         t1.end_node_id,
         t3.node_name,
         t3.node_type,
         t4."CalcMethode",
         t4."Speed",
         t4."speedTypeID",
                  round (((SDO_GEOM.SDO_LENGTH(t2.GEOMETRY3D,0.5, 'unit=M'))/2))
         ,
         CASE t4."CalcMethode"
         WHEN 1 THEN   (((SDO_GEOM.SDO_LENGTH(t2.GEOMETRY3D,0.5, 'unit=M'))/2) * t4."Speed")
        WHEN 2 THEN t4."Speed"
         ELSE 0
        END,
        t2.GEOMETRY3D  -- this is the geometry of the start_NodeSurface


  FROM   linksurface t1,
         nodesurface t2,
         nodesurface t3,
         "SpeedMatrix" t4
  WHERE  t1.start_node_id = t2.node_id
         AND t1.end_node_id = t3.node_id
         AND t4."accesTypeID" = t1.link_level;
```

2) PATHVIEW

```
  CREATE  OR  REPLACE  FORCE  VIEW  "PATHVIEW"  ("PATH_ID",  "PATH_NAME",  "SEQ_NO",  "LINK_ID",
"LINK_NAME", "Distance", "Timeduration", "GEOMETRY3D") AS
  select  T1.path_id  ,  T1.Path_name  ,   t2.seq_no,  T3.link_id,  T3.link_name,  t3."Distance",
round(T3."Timeduration")  , T3.GEOMETRY3D

     from route T1 inner join routeelement T2 on  T1.path_id = T2.path_id
     inner join "LINKSpeedVIEW" T3 on t3.link_id= t2.link_id and t3."speedTypeID"= t1.path_type
     order by t1.path_id, T2.seq_no;
```

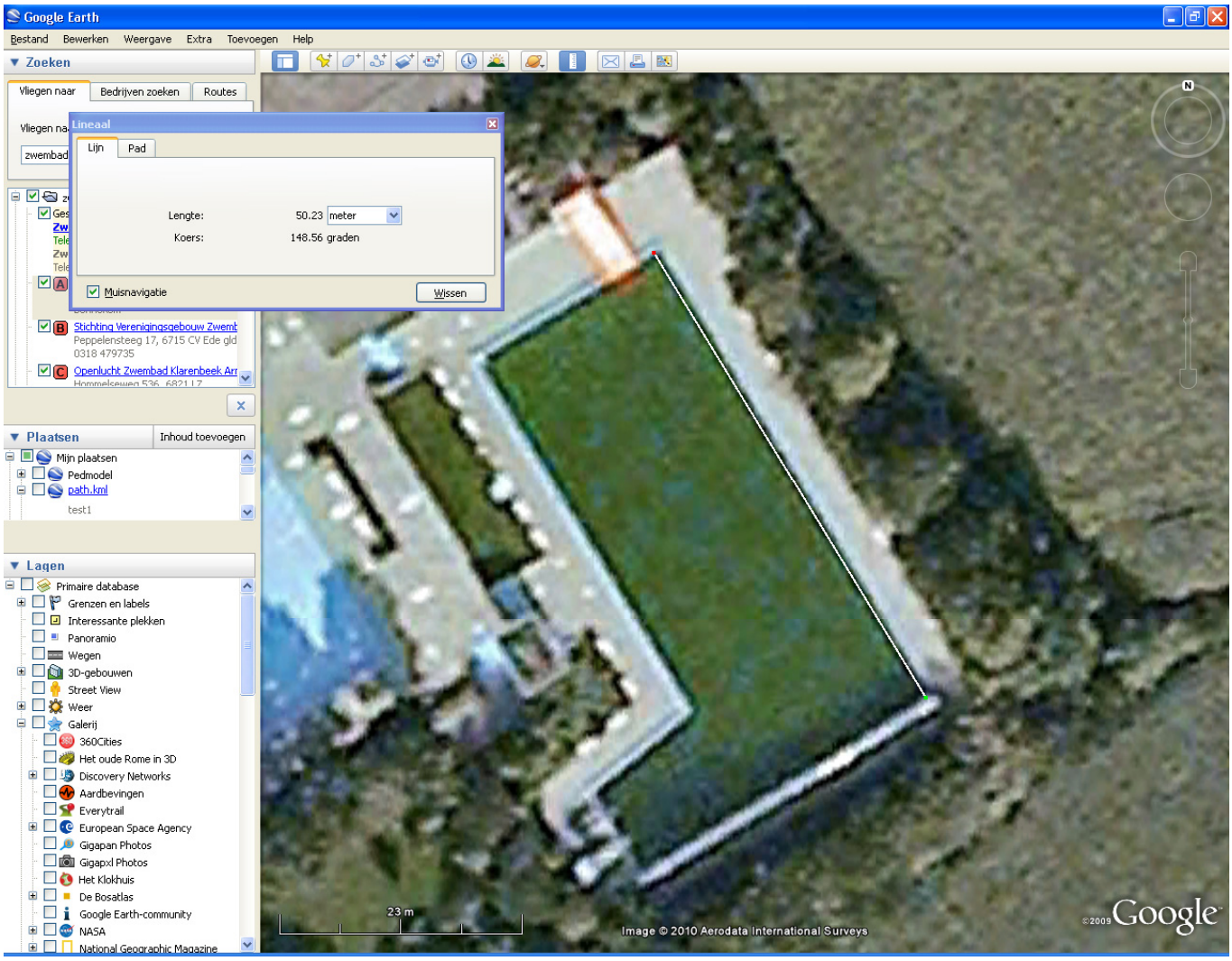
3) PATHDETAILVIEW

```
CREATE OR REPLACE FORCE VIEW "PATHDETAILVIEW" ("PATH_ID", "LINKSURFACELIST") AS
  SELECT
Path_ID,
```

```
SUBSTR(MAX(REPLACE(
SYS_CONNECT_BY_PATH( link_id  , '/')
,'/',' ')),2) Linksurfacelist
FROM (
select
path_id, seq_no , link_id,
row_number() OVER (Partition by Path_ID order by seq_no ) ROW#
from routeelement )
START WITH ROW#=1
CONNECT BY PRIOR  path_id = path_id
AND PRIOR row# = row# -1
GROUP BY  path_id;
```

4) TESTRESULTS

```
CREATE OR REPLACE FORCE VIEW "TESTRESULTS" ("PATH_ID", "PATH_NAME", "LINKSURFACELIST",
"Duration") AS
  select  T1.PATH_ID, T1.patH_NAME,  T2.linksurfacelist ,round(T1.COST)  FROM ROUTE T1 ,
pathdetailview T2
  WHERE T1.path_id= T2.path_id  ORDER BY path_type, T1.path_name;
```

P. Google Line tool

A measure was done of an object of know length (50 meter pool) nearby the studied area to check the usefulness of the Google line tool. (paragraph 7.1)