Dissertation

WAYFINDING IN GIS: FORMALIZATION OF BASIC NEEDS OF A PASSENGER WHEN USING PUBLIC TRANSPORTATION

ausgefürt zum Zwecke der Erlangung des akademischen Grades eines

Doktors der technischen Wissenschaftenunter der Leitung von

O.Univ.Prof. Dipl.-Ing. Dr.techn. Andreas Frank

E127

Institut für Geoinformation und Kartographie

eingereicht an der Technischen Universität Wien

Fakultät für Mathematik und Geoinformation

von

Dipl. Atm. Sc. Elissavet Pontikakis

Matrikelnummer 9927319

Hameaustrasse 14/5

1190 Wien

Wien, July 2006

Dissertation

WAYFINDING IN GIS: FORMALIZATION OF BASIC NEEDS OF A PASSENGER WHEN USING PUBLIC TRANSPORTATION

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Technical Sciences

Submitted to the Vienna University of Technology

Faculty of Mathematics and Geoinformation

By

Dipl. Atm. Sc. Elissavet Pontikakis

Hameaustrasse 14/5

1190 Wien

Advisory Committe:

Univ. Prof. Dipl.-Ing. Dr.techn. Andrew Frank

Institute for Geoinformation and Cartography

Vienna University of Technology

Univ. Prof. Dipl.-Ing. Dr.techn. Werner Kuhn

Institute for Geoinformatics

University of Muenster, Germany

Vienna, July 2006

ABSTRACT

Humans can move in space by foot or are transported with the use of a vehicle. Moving in space is a physical and cognitive process. Physical movement refers to the change in location of the human body. The cognitive process refers to perceiving, learning and reasoning. As a result of cognition, humans sense the environment, learn, plan, acquire information, make decisions and implement their decisions with actions. Significant research is dedicated to pedestrian wayfinding and car navigation in regards to learning, mental representation, and information acquisition. The research on wayfinding with public transport focuses primarily on stochastic modeling of passenger behavior and movements.

A passenger's move with the use of public transport is a spatial wayfinding task which integrates a business component. When a passenger moves with public transportation, he performs business transactions in addition to spatial wayfinding tasks. Successful implementation of the business process establishes the legitimation of a passenger. *When considering the needs of a passenger for information, it is important to consider both the spatial and the business aspect of the wayfinding with public transportation because if any of the two aspects is missing the trip fails.*

This research simulates the passenger as a cognitive agent who moves in a space where different states occur for the agent. The agent transitions between states by performing operations. The agent creates a plan about his trip in advance. When he executes the trip, he perceives the environment and uses his plan to transition from one state to another. He perceives the affordances, and utilizes the knowledge stored in the world in meaningful structures and in signs to acquire information. He then uses his plan to make a decision and subsequently to perform an operation. Affordances are indicative to possible operations.

Ontology assists in defining the structure of a trip into processes and units of operations. Certain units of operations are repeated throughout a trip. Ontology also assists in identifying the level of detail for information needs during the different phases of a trip. Simulation with an agent provides a tool for modeling a concrete case of wayfinding with public transportation within the city of Vienna, Austria. The formal model is based on algebraic specifications of the operations involved in the overall task. The model can be used to develop practical methods for assessing the information provided to passengers during a trip with public transport.

There are three scientific contributions from this research. The first scientific contribution is the integration of the spatial wayfinding and business process into the overall wayfinding task. The second contribution is the identification of the structure of objects and operations and their connection to the level of detail and the phases of the trip execution. The third contribution is the identification of the minimum information needs required to complete a trip with public transportation.

KEYWORDS

Public Transportation, Wayfinding, Business Process, Ticketing, Affordances, Algebras, Simulation, Passenger, Traveler, Agent, Cognition, Operations, Information

KURZFASSUNG

Personen können räumlich sowohl zu Fuß als auch mit einem Fahrzeug fortbewegen. Diese räumliche Bewegung einer Person konstituiert einen physikalischen und einen kognitiven Prozess, wobei der physkalische lediglich die Änderung des Aufenthaltortes der betrachteten Person beschreibt. Der kognitive Prozess besteht aus Wahrnehmen, Lernen und Schlussfolgern seitens der Person während ihres Ortswechsels. Durch Kognition nehmen Menschen ihre Umwelt wahr, lernen, planen, beschaffen Informationen, treffen Entscheidungen und setzen ihre Entscheidungen in Handlungen um. Dem Bereich der Fußgänger- und Fahrzeugnavigation sind signifikante Forschungsarbeiten mit Fokus auf Lernen, mentaler Repräsentation und Informationsbeschaffung, gewidmet worden. Im Bereich Wegesuche mit öffentlichen Verkehrsmitteln konzentriert sich die Forschung auf stochastische Modelle, die das Verhalten und die Fortbewegung von Fahrgästen repräsentieren.

Die Fortbewegung eines Fahrgastes mit öffentlichen Verkehrsmitteln ist eine räumliche Wegesucheaufgabe, die zusätzlich einen geschäftlichen Prozess beinhaltet. Die erfolgreiche Umsetzung dieses Prozesses führt zur Legitimierung des Fahrgastes. *Betrachtet man den Gesamtinformationsbedarf eines Fahrgastes für eine erfolgreiche Wegesuche, so ist es wichtig, sowohl den räumlichen als auch den geschäftlichen Aspekt der Wegesuche mit öffentlichen Verkehrsmitteln zu beachten, da die Wegesuche fehlschlägt, wenn irgendeiner der beiden Aspekte fehlt.*

Die vorliegende Forschungsarbeit simuliert einen Fahrgast als kognitiven Agenten , die sich im Raum bewegt und mit verschiedenen Situationen konfrontiert wird. Durch das ausführen von Operationen wechselt der Agent zwischen verschieden Zuständen. Bevor der Agent seine Reise antritt, plant er sie. Während der Fahrt nimmt er seine Umwelt wahr und benützt seinen Plan um von einem Zustand in den anderen zu wechseln. Er nimmt Affordanzen die mögliche Operationen anzeigen, wahr, und verwendet das Wissen, das in der Welt in verständlichen Strukturen und Zeichen gespeichert ist, um Informationen zu akquirieren. Er benutzt daraufhin seinen Plan um Entscheidungen zu treffen und diese in Operationen umzusetzen.

Ontologien werden verwendet um die Struktur einer Fahrt in Prozesse und Operationseinheiten zu unterteilen. Gewisse Operationseinheiten werden während der Fahrt wiederholt. Ontologien unterstützen auch die Bestimmung des notwendigen Detailgrades der benötigten Informationen während der verschiedenen Phasen der Fahrt. Die agentenbasierte Simulation stellt ein Werkzeug für die Modellierung einer konkreten Fallstudie für eine Wegesuche mit öffentlichen Verkehrsmitteln in der Stadt Wien, Österreich, zur Verfügung. Das formale Modell basiert auf algebraischen Spezifikationen der Operationen, die in einer Wegesucheaufgabe vorkommen. Das Modell kann für die Entwicklung praktischer Methoden zur Bestimmung und Bewertung der Informationen, die Reisende während einer Fahrt mit öffentlichen Verkehrsmitteln bekommen, verwendet werden.

Zu dieser Arbeit gibt es drei wissenschaftliche Beiträge. Der erste Beitrag beschreibt die Integration der räumlichen Wegesuche und des Geschäftsprozesses in die gesamte Wegesuchaufgabe. Im zweiten Beitrag werden die Objektstrukturen und –operationen und Ihre Verbindung zum Detailierungsgrad und den Reisephasen aufgezeigt. Im dritten Beitrag wird schliesslich der Informationsbedarf bestimmt, welcher zumindest erforderliche ist, um die Reise erfolgreich abzuschliessen.

SCHLAGWORTE

Öffentliche Verkehrsmittel, Wegesuche, Geschäftsprozess, Kartenverkauf, Affordanzen, Algebren, Simulation, Fahrgast, Reisender, Agent, Kognition, Operationen, Information

ACKNOWLEDGEMENTS

Acquiring a PhD is not only getting a title, a publication, and maybe a new ranking in the job market. It is mostly obtaining a lifetime experience which shapes the way you think and act from that time onwards. This work gave me the opportunity to learn new topics and to revisit older ones. It helped me to further appreciate the great thinkers of all times who paved the road to reasoning. I am vastly grateful for their existence.

As to specific individuals, I am grateful to Professor Andrew Frank that honored me by guiding me through this research with patience and understanding. I have acquired a great many things from our long discussions which I will treasure for the rest of my life. I am also thankful to my second advisor Werner Kuhn who provided me with constructive comments and recommendations for improvements.

I thank all my colleges at the Institute for Geoinformation and Cartography for their support. I especially thank Claudia Achatschitz and Barbara Hofer for their never ending assistance with the German language and Florian Twaroch for our discussions on cognitive agents and formal models. I am grateful to Edith Unterweger not only for her administrative assistance but also for her faith on me and her encouragement.

I am thankful to my husband John Coyne for his patience, his time when listening to cognitive public transport issues, totally out of his area of interest, and his confidence in me. I thank him for his assistance in technical issues and text editing. I would not have made it without the support and motivation of my most cherished girls: Kassanthra and Ariathney who at all times demonstrated an immense trust in my abilities.

I would always go around preaching that the best period in someone's life to perform a PhD research is before a family of more than two is concealed or after this family is dispersed into individual celestial bodies. I can say now that every time is suitable for a PhD, each one for each own good reasons. In my case, the fact that I could become a role model to my young girls was the greatest motivation to go on despite the difficulties and set backs.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This chapter provides the motivation for this research and describes the underlying research question. The hypothesis is that it is possible to identify an explicit set of information needs that a passenger employs while executing a door-to-door trip with the use of public transportation and that it is possible to identify these information needs with the use of agent-based cognitive modeling. The target audience for the thesis is identified and the expected scientific results and probable future uses of these results are discussed. Finally, this chapter provides an overview of the structure of this thesis and of the content of each chapter.

1.1 Motivation

Automobiles are blamed for a number of evils that are associated with city life ranging from respiratory problems to the global green house effect. Public transportation is considered an ecological approach for moving around in today's cities and it is widely promoted in cities. Significant efforts focus on attracting passengers to use public transportation in the USA. Elsewhere, including many European cities, a lot of manpower and technical resources are utilized by transportation authorities to expand and improve their services for existing users, to attract new users and to respond for increased demand in the future.

Significant steps have been taken in improving the physical aspect of public transportation infrastructure around the world with safer and faster vehicles, smoother rides etc. The field, however, is still open to research and innovation in order to optimize the information aspects of services offered by public transportation providers. This research focuses on the identification of the minimum set of information needs of a passenger who performs a spatial and business wayfinding task when moving from point A to point B in a city using public transportation.

When a traveler conducts a trip with public transportation, he performs a spatial wayfinding task. This task is defined as going from one point to another, moving in space.

The use of public transportation implies the use of payment within a business model. The passenger pays for his trip and receives a ticket as a proof of payment for transport. This makes the passenger a legitimate passenger, which is a requirement for the use of public transportation. A passenger is illegitimate when he does not have a ticket or when his ticket is invalid, which can happen for different reasons. For example, a prospective passenger may not have enough money, the correct currency, or the correct denomination to pay for the ticket. In this case, he can still go from one point to another, but not as a legitimate passenger with public transportation. *This research introduces the business process in the overall wayfinding task.*

During a trip, a passenger performs several spatial wayfinding tasks such as finding the bus stop or finding the ticket vendor. The passenger also has several legitimation related transactions such as buying and validating the tickets, especially in multi-modal trips. Operations performed during the wayfinding process present similarities with operations performed during the business processes. The operation of getting off a transportation vehicle is part of the wayfinding process of a trip, and moves a passenger from inside to outside a vehicle. The vehicle is an object with a content and a capacity. When a passenger pays for the ticket he moves money from his pocket to the ticket counter. The passenger's pocket has a content and a capacity. The vehicle and the passenger's pocket are referred to as containers in this research.

People first conceive and decide their trip. They then make a plan about the trip, and finally they execute their travel. During the conception phase, they need general information about the origin and the destination of their trip, such as: go from the Eiffel Tower to the Arch de Triumph in Paris. In the planning phase, more detailed information is needed such as: "what vehicles can be used to go from the Eiffel Tower to the Arch de Triumph?" and "where are the vehicle stops?" Also in this phase, people inquire about the need for a ticket. When they physically make the trip, they need more detailed information such as: "what is the way to the vehicle stop?", "where can one buy a ticket for the vehicle?", and "what form of money can be used to pay for the ticket?" *Information at different levels of detail is needed in each phase, for both the spatial and the business aspect of the trip.*

The information collected before the trip is stored in a plan which represents the traveler's mental plan of the intended trip. The mental plan assists the traveler to execute the trip. On the other hand, there is information stored in the environment which also assists the traveler to execute his trip. This information is referred to as knowledge in the world. This knowledge, for example, assists the traveler in distinguishing between a bus and a garbage truck or between a coin and a bank note, and helps him to use a vending machine. *The information collected during planning, and the knowledge in the world assist the traveler in executing the spatial wayfinding and business processes in his trip.*

A transportation system is often judged by the time it takes for a passenger to travel between two fixed points. It is evident that this approach overlooks a large amount of time and effort which is attributed to information acquisition, retrieval, decision making and to the execution of the financial transactions for ticketing. For the inexperienced traveler a significant amount of time is spent in acquiring information about a prospective trip relevant to schedule, ticketing rules and fares, instructions on the operation of ticket vending and validation machines, and effective navigation signage, etc. A similar situation occurs when a traveler arranges and flies by airplane from one point to another. The amount of time required for flight arrangement, decision making and booking is comparable to the time between take off and landing. It is not unusual for a passenger to spend more time acquiring information about the trip and executing the ticketing transactions than to physically move from point A to point B.

The term information is used in connection to the knowledge needed for decisions (Edwards 1967). In this thesis, the content of information is equivalent to data and is uniformly interpreted by all travelers. It expresses the same pragmatic measure (Frank 2003) thus a bus stop for example is perceived and interpreted as a bus stop by all travelers. In this sense, the information needs is a list of data needed to perform the overall wayfinding task.

This thesis uses the term traveler when referring to a person who changes his location and the term passenger when referring to a person which is been transported in a vehicle operated by someone else. In this context, a person involved in a wayfinding task who uses a vehicle for part of the way is referred to as a traveler. When this person is in a public transportation vehicle or waits at a bus stop to board a vehicle he is referred to as a passenger.

1.2 Goal and hypothesis

This research aims to assess the information needs for both the business transactions and physical movements of a passenger who uses public transportation for going from one point to another. The aim is to provide answer to the following questions in reference to the spatial wayfinding and business needs of a passenger:

1. Is it possible to systematically identify the minimum information needed by a passenger when using public transportation?
2. What information is embedded in the environment, perceived, and subsequently interpreted by a passenger who moves with public transport?

Our hypothesis is that:

1. *There is minimum information needed by a passenger who executes a wayfinding task with public transport.*
2. *Simulation will reveal the information needs for the spatial wayfinding and business aspects of moving from point A to point B with public transport. The amount of information needed is linked with the level of detail employed during the simulation.*
3. *The operations involved in the spatial movements and business transactions of a passenger during a single-mode trip with public transportation form ontological units of operations. A trip where more than one mode of transportation is involved is accomplished with the iteration of these ontological units of operations.*

1.3 Approach and research background

The passenger's overall task is wayfinding in the spatial and business realm. The spatial aspect of the wayfinding task has been previously modeled by Raubal (Raubal 2001). The novelty of the current research lays in the incorporation of the business component in the overall wayfinding task.

The scenarios which motivated this research are initially examined. In the first scenario a businessman travels from the airport to his hotel in the center of Vienna, Austria. In the second scenario a family travels to a mountain for the weekend to go hiking. In both scenarios, the trips are executed using public transportation. This research examines the questions the travelers need to answer at the transfer points between transportation modes, which are the points where decisions are taken. This research focuses on the information needed to answer these questions and make the decisions. The decision process which goes into effect at the decision points is beyond the scope of this research.

Due to the complexity of the motivating scenarios, a reduced case is used to construct the conceptual scheme. The conceptual scheme is formalized with the use of agent theory (Ferber 1998). The agent is structured to plan his trip, to perceive the environment, and to act upon the environment. The use of ontology and algebraic specifications lead to the abstract formal model (Frank 2005-Draft) of wayfinding with public transport.

This research uses the previous work on spatial wayfinding (Raubal 2001) and models the agent's behavior during the spatial and business processes of the wayfinding task with public transport. This is performed with the use of the theory of perception and knowledge which is embedded into the world in the form of affordances and image schemata (Lakoff and Johnson 1980; Gibson 1986; Norman 1988). The theory of signs is also considered (Dewey 1938; Pierce 1992).

1.4 Target audience

This work addresses the following disciplines:

- Transportation analysts and transportation modelers can use this research to assess the information which is provided to passengers by transportation providers.

- Agent-based modelers in the area of artificial intelligent can apply this research in modeling the passengers' interaction with artifacts related to public transportation such as time tables, ticket vending machines etc.
- Designers of transportation related artifacts can use this research as a means for assessing the use of an artifact in a specific place.
- Computer scientists working on application of Geographic Information Systems (GIS) and location-based services can benefit from the formal structure and specifications of this work.
- Cognitive scientists and physiologists can use this research to model the behavior of passengers using a public transportation system.
- Philosophers can apply the ontological classification constructed for this research to examine the hierarchies of operations in other domains.

1.5 Structure of this thesis

Chapter 2 presents the two scenarios which motivated this research and the reduced case. It describes three phases for each scenarios which correspond to the conception, planning and execution of each trip. The chapter introduces the concept of hierarchies in the structure of operations performed by travelers, and the concept of the traveler's mental plan.

Chapter 3 describes the theories which support the conceptual and formal model. It starts by examining multi-agent systems. Cognitive agent theory is used to explain the passenger's behavior as he perceives his environment and interacts with it. Affordances are embedded in the environment which are perceived and interpreted by the agent. The chapter discusses the knowledge in the world, the affordances and image schemata. It also discusses the theory of signs. Subsequently, this chapter presents the literature of spatial wayfinding. Finally it concludes with the system's epistemology and ontology(Frank 2001).

Chapter 4 presents the conceptual model derived from the reduced case. It describes the components which constitute the conceptual model namely, the processes and operations, the passenger, the ticketing, the vehicle, and the point network. This chapter presents the objects and the operations which structure the conceptual model and provides a connection with the different levels of detail as well as the different phases of the trip execution.

Chapter 5 presents the data structures, the state transition diagram and the entity relationship. It introduces the formation of algebras for representation of real world phenomena, and functional programming languages. The tools discussed in this chapter are used to formally structure the conceptual scheme.

Chapter 6 describes the ontological decomposition of a traveler's multi-modal trip into processes and operations. It presents the spatial wayfinding and business processes and the different levels of operations. This chapter explains the agent's transition from one state to another and provides the connection between the questions at each point of the point network and the information needs. The chapter concludes with the identification of the minimum information needed by the agent to complete his trip legitimately with public transportation.

Chapter 7 describes the components of the formal model used in this research. The formalization conforms to the structure of operations and is an adaptation of the algebraic specifications. This chapter explains the encoding of the state transitions and the relation model with the use of functional programming.

Chapter 8 summarizes the work presented in this thesis. It describes the findings of this research and discusses future directions. Appendices I and II provide the Haskell encoding used in the formalization.

1.6 Text quotes

There is a quote at the beginning of each chapter which is sometimes humorous and other times serious. Scenes inspired by, or taking place within, the world of public transportation are often encountered in literature. Much can be learned by examining these scenes. Literature, predominantly fiction, is rich in examples of moving busses, trams, trolleys, gondolas, horse wagons and many other vehicles supported by some type of infrastructure, and used by all types of passengers. Hall suggests that an author looks closely into the scenes which he describes and a scientist can, therefore, utilize in turns these views. He states, " What would be the result if, instead of regarding the author's images as literary conventions, we were to examine them very closely as highly patterned reminder systems which release memories?" (Hall 1966). The author of this thesis appreciates Hall's view and includes the quotes at the beginning of each chapter as a source for contemplation and a humorous connection to the world of public transportation.

2. CASE STUDIES OF WAYFINDING WITH PUBLIC TRANSPORTATION

> *"... 'Ay, and here they come to look at our tickets,' said the banker, fumbling in his clothes. The lamps were paling in the dawn when the halfcaste guard came round. Ticket-collecting is a slow business in the East, where people secret their tickets in all sorts of curious places. Kim produced his and was told to get out. 'But I go to Umballa,' he protested. 'I go with this holy man.' 'Thou canst go to Jehannum for aught I care. This ticket is only to Amritzar. Out!' ..."*
>
> *Kipling-Kim, p316*

In the above quote, Kipling provides an amusing description of a scene in a public transportation vehicle, which demonstrates the difficulty of having the right ticket. Although Kim was first published at the beginning of the 20^{th} century, travelers today still face difficulties when confronted with issues related to moving with public transportation as this chapter will demonstrate.

This chapter describes the two motivating real world scenarios and provides initial insights of this research. It also presents the reduced case scenario which leads to the development of the conceptual model described in Chapter 3. The two motivating scenarios were originally designed and carried out for the Bundesministerium für Wissenschaft und Verkehr in Vienna, Austria as part of its effort to identify missing components in the information chain for public transportation. The project focused on the door-to-door analysis of two multi-modal scenarios where the transportation space is shared between passengers and transportation vehicles.

In the first scenario, a businessman goes from Schwechat Airport in Vienna to his hotel in the center of the city. His employer pays for all transportation related expenses from the airport to his hotel, thus the traveler has no financial concerns. The traveler uses public transportation because he trusts the high reputation of the local public transportation network. The businessman arrives on a Saturday and his business meeting starts on Monday, thus, there is no time pressure during his trip from the airport to his hotel. The traveler is an English speaking, non-handicapped individual with average learning and understanding abilities.

In the second scenario, a family of four from Vienna is heading for a recreational weekend trip to a mountain in Steirmark. The family is composed of non-handicapped English speaking individuals with average learning and understanding abilities and some knowledge of the German language. In Vienna, many schools operate on Saturday and the children involved in this scenario attend Saturday school. The family leaves after school finishes on Saturday. The family does not own a private car, and the two adults and two children travel using multi-modal public transportation. The financial aspect is important in this trip as this scenario moves four members of a middle income family. The travelers have a time limitation in order to reach the last scheduled lift of the day to the mountain.

2.1 Phases of a trip implementation

Each trip is first conceived, then planned and finally executed. Literature agrees in analyzing trips in planning and execution phases (Werthner 2003). Each phase is associated with a different level of detail in the information needed by the traveler. As a trip evolves from the conception to the planning through to the execution phase, the information needed ranges from coarse, to medium, to fine level of detail respectively (Timpf, Volta et al. 1992).

In the conception phase, the traveler decides the trip and identifies the starting and ending point of the trip. He decides on the time of his trip and the intended mode of transport. This is referred to as the coarsest level of information detail concerning the trip. In the planning phase, the traveler identifies the areas where a change in the mode of transport occurs. He identifies the destination of each mode and the schedule. This is a personal judgment by the traveler and depends on 1) objective factors such as the frequency of a transportation mode and the overall duration of the trip, 2) subjective factors such as importance of subsequent activities and impediments due to external conditions such as the weather, and 3) emotional state. In the planning phase, the traveler identifies or confirms the need for payment. This information conforms to the traveler's experience to pay for goods and services. The level of detail in information needed during the planning phase is referred to as medium. In the execution phase, the traveler needs a fine level of detail in information such as where the bus stop is, where the ticket vending machine is, what denomination he can use for his payment etc. He acquires this information on the spot while involved in the task.

For each scenario, the information needed by the traveler in the conception and planning phase is provided at the beginning. In a multi-modal trip, information involving a subsequent mode of transport can be obtained while in the previous mode or at the point of mode change. We consider this information as information needed for the planning phase of the succeeding mode and thus as information needed in the planning phase of the overall tip. Subsequently, the execution of the trips is described in a narrative form. This narration provides links to the information collected in the conception and planning phases and demonstrates the information needs during the execution phase. Information collected in the conception and planning phases represents the trip plan stored in the traveler's mind. This collection of information includes also knowledge obtained from experience as the scenarios demonstrate. The scenarios are described in a somewhat standardized language as a first step towards formalization. Time is not considered except in the cases where there is a time conflict.

2.2 Description of scenario 1

Title: A businessman travels from the Vienna Schwechat Airport to the Hotel Sacher in the center of Vienna.

2.2.1 Conception phase

In this scenario, the traveler decides to travel from Vienna Schwechat Airport to the

Hotel Sacher in the center of Vienna. The intended time of his trip is Saturday morning and his mode of transport is public transportation. Figure 2-1 shows the location of Vienna's Schwechat airport and the Hotel Sacher. Number (1) indicates the general area of the origin while number (4) the general area of the destination.

2.2.2 *Planning phase*

The traveler identifies the bus and the metro as his modes of transport. The numbers (2) and (3) on Figure 2-1 indicate locations where a change in the mode of transport occurs. In this phase, the traveler identifies the destination of each mode a) bus to the City Air Terminal, Hotel Hilton (2) and b) metro to Karlsplatz (3). The Sacher hotel is across the State Opera within walking distance from the metro stop "Karlsplatz". The need for payment is identified by experience in this phase and the traveler comes prepared with local currency.

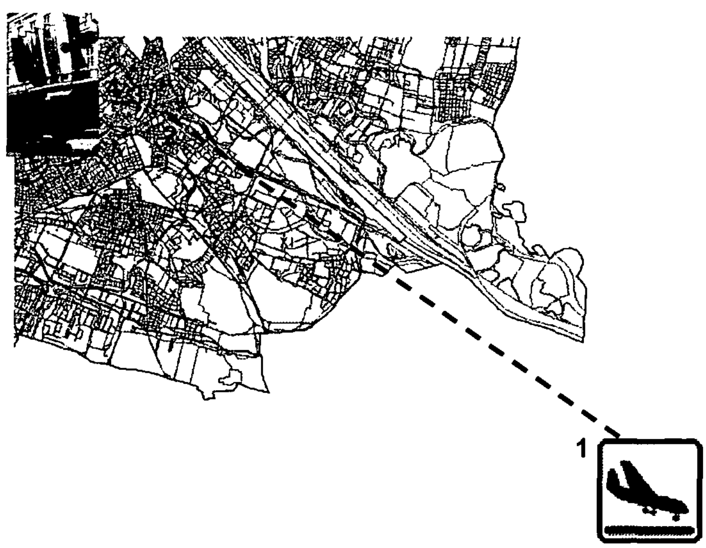


Figure 2-1. Location of Vienna's Schwechat airport (1), Hotel Sacher (4) and intermediate stops (2) and (3) for scenario 1

2.2.3 *Execution phase*

The traveler's goal is to find Hotel Sacher. We break down the description of the trip for each one of the areas where the traveler needs to make a decision concerning the mode of transport. We call these areas "decision points for selecting a wayfinding mode" as they refer to points on Figure 2-1.

Decision point 1: The airport

In the conception phase the traveler identifies that the Hotel Sacher is located in the center of Vienna across from the State Opera. At the airport, he finds out which available mode of transport will take him to the center. The information desk at the airport is closed at this time. The traveler looks at the signs, recognizes a sign for buses and he decides to use the bus as his first mode of transport. His choice conforms to the *"Individual Choice Behavior"* of the *"ubiquitous independence of irrelevant alternatives"* and that of the experienced rewarded response choice theories (Luce 1967). Luce supports that there is the same probability for someone to make a choice which he has made before, regardless of the increase in the number of new alternative selections. He also suggests that when a subject makes a rewarded choice the probability of making the same choice again increases (Luce 1967).

The traveler needs to identify the bus destination. This task is performed by consulting a map, the Internet or a tourist guide or by asking someone else. In this case, the user consults a paper map and takes a decision between the following three bus destinations: a) the City Air Terminal, Hotel Hilton, b) the Südbahnhof train station or c) the Westbahnhof train station. The common map of Vienna, which is provided by the tourist or information service agencies free of charge, shows the City Air Terminal on the Public transportation lines map section (Öffentlich Verkehrsmittel). The City Air Terminal is shown on the map to be closer to the center than the other two bus destinations. Based on the proximity between his final destination and the three bus destinations, the traveler selects the bus which goes to the City Air Terminal, Hotel Hilton. He then follows the signs and walks to the bus stop designated by a yellow bus stop sign.

The bus is at the bus stop. The passenger knows from experience that transportation is a service that requires payment and he inquires information about the ticket from the bus driver. In Austria, tickets can be purchased from vending machines, tobacco stores or from the bus driver inside the bus (Winter, Pontikakis et al. 2001) with local currency. The traveler has local currency, boards the bus and buys a one-way ticket from the bus driver.

Buying the ticket from the driver also validates the ticket and the traveler is now a legitimate traveler. The traveler retains his ticket. Figure 2-2 shows a typical bus which connects the airport to the Vienna city center. In this trip, the bus goes directly to its destination without intermediate stops. This is where all passengers leave the bus.

Figure 2-2. First mode of transport in scenario 1.

Decision point 2: City Air Terminal, Hotel Hilton

The traveler is now outside the bus. He looks at the map and he recognizes that he is still far away from his final destination. The map indicates that there is a metro line connecting this location with the State Opera. He follows the signs and walks to the adjacent metro station. Stationary artifacts such metro signs and dynamic artifacts such as the cluster of people entering and exiting the station assist him in the identification of the entrance of the metro station (Pontikakis 2005).

The traveler takes the conservative approach that the ticket obtained earlier for the bus from the airport is no longer valid and he looks for the ticket vending machines (Phillips and Edwards 1967). He finds a ticket vending machine and follows the instructions. The traveler does not understand the option of zoning which refers to the area where someone can travel with the use of a ticket. The traveler selects the Zone One ticket as it is presented as a default option in the vending machine (Payne, Bettman et al. 1993; Dhar 1997). He selects a one-way ticket and pays with local currency. The traveler has the required denomination for this transaction. Stationary artifacts and the design of the ticket itself indicate the need for ticket validation. The traveler sees that there are validation artifacts in the vicinity and that the ticket has a blank space and an arrow indicative to validation stamp (Pontikakis 2005) (Norman 1988). Thus, he validates and holds on to his ticket.

Subsequently, the traveler locates a metro map and identifies the metro line, direction, and exit stop. The passenger concludes to take the U4 (green line) in the direction to Hütteldorf (Figure 2-3) and exit at the Karlsplatz stop. He ignores the line U3 (orange line). He walks following the signs for the platform to U4 direction Hütteldorf. He waits at the platform and enters the next metro labeled Hütteldorf.

Figure 2-3. Second mode of transport in scenario 1.

In the metro, the traveler identifies the stop for exiting by listening to the loudspeaker announcements and watching the signs of each passing metro stop. He exits when he hears the loudspeaker announcement and sees the signs for Karlsplatz.

Decision point 3: Karlsplatz

The passenger is now on the metro platform. Based on his general knowledge about the destination, the passenger decides to walk to the Hotel Sacher. Figure 2-4 depicts the location of the Hotel Sacher and the Opera. He follows the signs for the Opera in the underground passage until he reaches the escalator to the street level. He goes up the escalator and sees the State Opera. Subsequently, he walks to the back of the Opera and sees the entrance to the Sacher hotel.

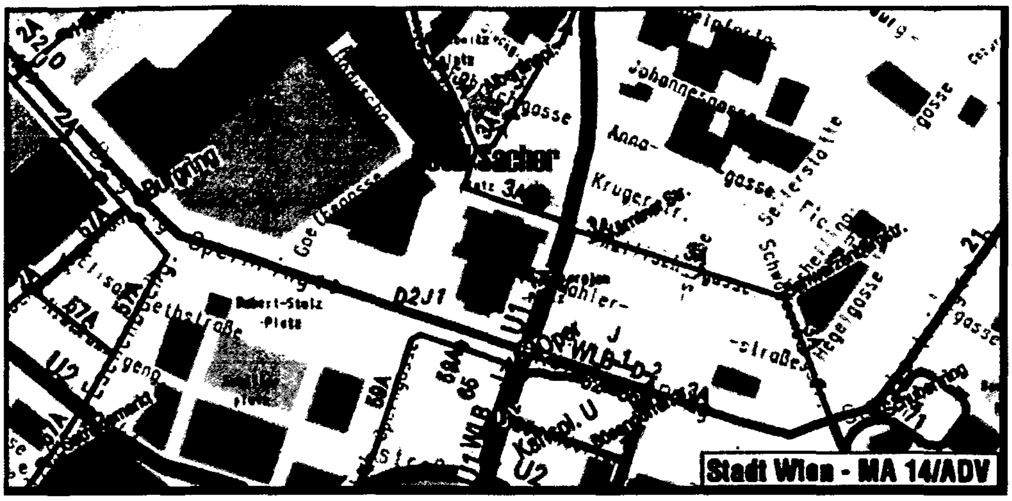


Figure 2-4. Location of the Hotel Sacher in the center of Vienna (*Source: Wien Grafik*)

2.3 Description of scenario 2

Title: A family of two adults and two children from Vienna travels for a weekend to Aflenz Bürgeralm to go hiking. Bürgeralm is a mountain in the Steirmark (Styria) region of Austria.

2.3.1 Conception phase

In this scenario, the travelers decide to take a trip from Vienna to Bürgeralm in Steirmark. In this phase they book a night at the hotel Naturfreunde on Bürgeralm and acquire the information that Bürgeralm is near Aflenz. The decided time for the trip is Saturday around noon and the intended mode of transport is public transportation. Figure 2-5 depicts the country of Austria and its regions. The area of interest is marked by a dashed line. The general areas of origin and destination are marked by solid lines.

Source: http://austria-tourism.at/index.html.en

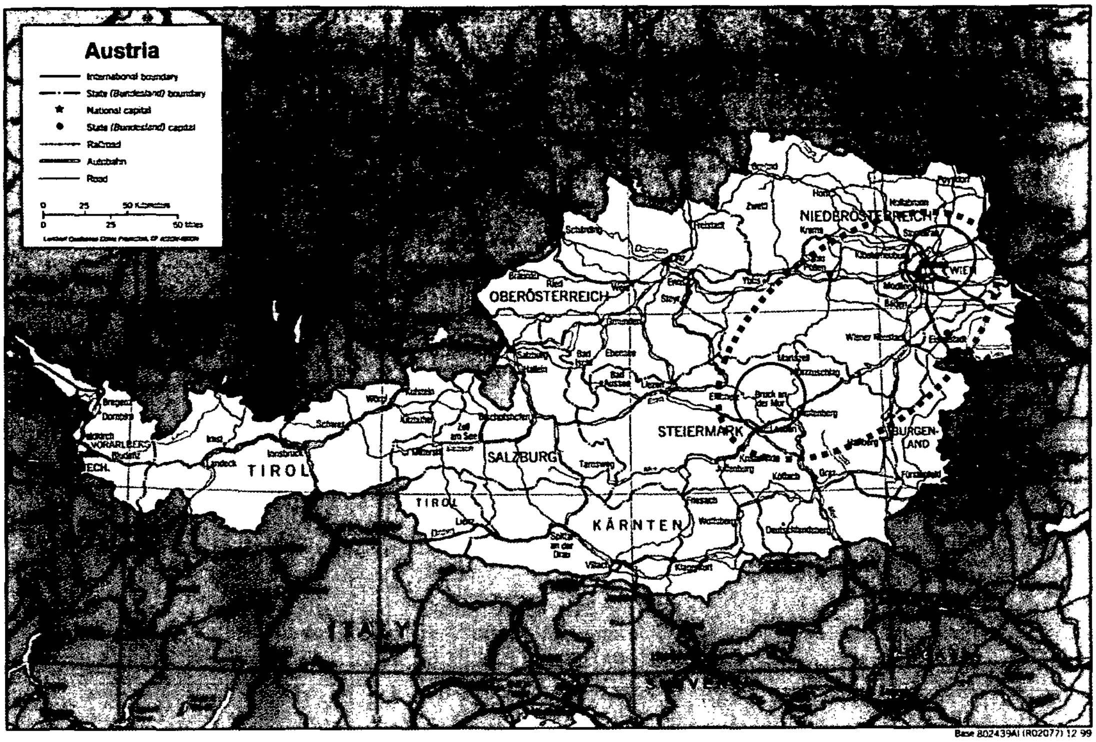


Figure 2-5. Regions of Austria

2.3.2 Planning phase

After talking on the phone, looking at maps and searching the Internet, the travelers identify the following transport modes and destinations: a) metro to Südtiroler Platz, b) train Bruck an der Mur, c) bus to Aflenz Kurort, d) lift to Bürgeralm. The hotel where they will stay is within walking distance from the lift end stop. The travelers identify the schedule of the modes involved in the trip and are prepared to pay for the trip. Figure 2-6 shows the location of Schwedenplatz in Vienna's center and Bürgeralm in Steirmark. The number (1) indicates the general area of the origin and the number (5) the general area of the destination. The numbers (2), (3) and (4) indicate areas where a change in the mode of transport occurs.

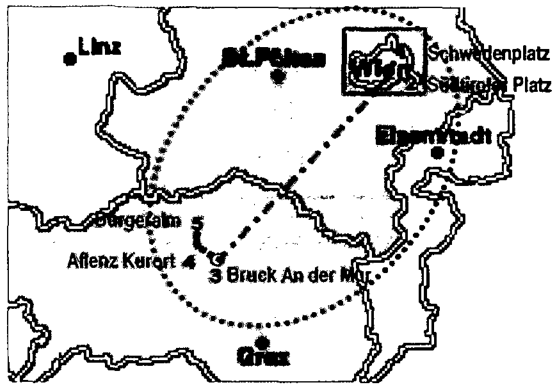


Figure 2-6. Location of origin (1), destination (5) and intermediate stops (2), (3), and (4) in scenario 2

2.3.3 Execution phase

The trip is described for each of the areas where the travelers need to take a decision concerning the mode of transport. These areas are referred to as "decision points for wayfinding mode" are shown on Figure 2-6.

Decision point 1: Schwedenplatz

The travelers are outside the children's school in Vienna's Schwedenplatz. They have previously looked at a map and know from it that the metro is the first mode of transport for their trip. The map indicates that there is a metro stop in Schwedenplatz and they walk finding their way to the entrance of the metro stop. Similar to scenario 1, stationary and dynamic artifacts assist them in the identification of the entrance of the metro station (Pontikakis 2005).

The passengers know from experience that the service for transportation requires payment and they look for the ticket vending machines. They find a ticket vending machine and follow the instructions. They select two one-way tickets for adults and two one-way tickets for children. As the concept of zoning is not clear to them, they select Zone One tickets as the simplest option and pay with local currency. The travelers realize then that they do not have the required denomination for this transaction and they run to the near-by bakery to make change. They return to the ticket vending machine, repeat the transaction, and obtain their tickets. Stationary artifacts and signs on the ticket itself indicate the need for ticket validation (Norman 1988) (Pontikakis 2005). The travelers see that the ticket has an area indicative to validation stamp and that there are validation artifacts in the vicinity. Thus, they validate and hold on to their tickets.

Subsequently, the travelers locate a metro map and identify the metro line, direction, and exit stop. The passengers conclude to take the U1 (red line) in the direction to Reumannplatz and exit at the Südtiroler Platz stop. They ignore the line U4 (green line). They walk following the signs for the platform to U1 direction Reumannplatz. They wait at the platform and enter the next metro labeled Reumannplatz. In the metro, the travelers identify the stop for exiting by counting four stops after entering (Figure 2-7). They confirm and exit when they hear the loudspeaker announcement and see the signs for Südtiroler Platz.

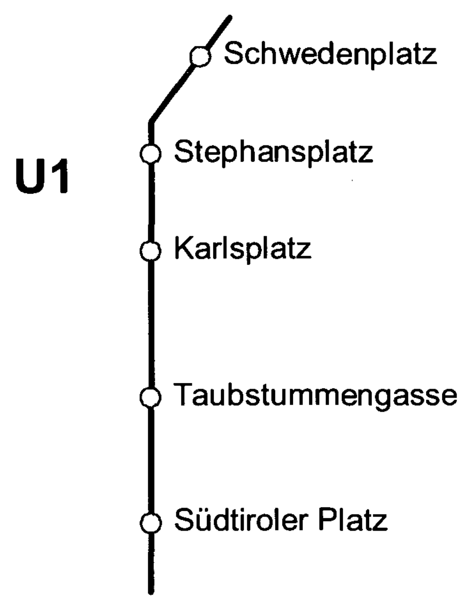


Figure 2-7. Metro stops between Schwedenplatz and Südtiroler Platz on U1

Decision point 2: Südtiroler Platz and Südbahnhof

The passengers are now at the metro platform at Südtiroler Platz. Figure 2-8 shows the interior of Südtiroler Platz metro station with the signage guiding to neighboring transit modes and to Südtiroler Platz. In the planning phase, the passengers have identified the train as the next mode of transport and the Südbahnhof train station as their boarding station.

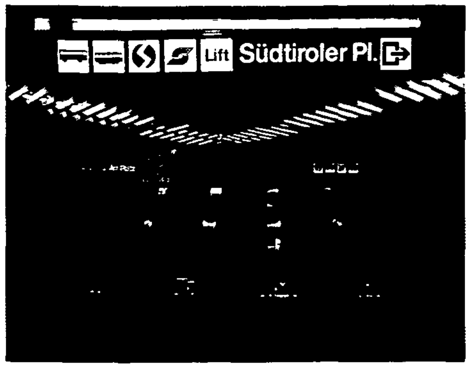


Figure 2-8. First mode of transport in scenario 2.

Subsequently the travelers search for signs for the train station; in their case, there are no signs they recognize for the train station and they choose among the following three options: a) inquiring information from fellow passengers; b) consulting a local map and take the exit that brings them to the street leading to the train station; and c) selecting an exit arbitrarily hoping that later signs will assist them to recover from any mistakes which they may have made in identifying their way to the train station. The travelers select the third option and proceed up the steps to the street. At the street level, they look for artifacts to assist their wayfinding effort. They mistake the train station parking garage sign for the sign to the station itself. In this case, the two directions coincide. The sign to the train station parking garage is the only sign that they can observe bearing the name "Südbahnhof". The dynamic artifact of the cluster of people entering the train station assist the travelers in identifying the entrance (Pontikakis 2005).

The travelers are now inside the train station. In the planning phase, they have identified that the train which takes them to Bruck an der Mur, the next decision point for selecting a wayfinding mode, is the train to Graz. They have also identified the departure time of the train. In the train station, they confirm the departure time of the pertinent train on the schedule board and identify the proper platform. As previously, the travelers know that they need a ticket to board the train. They identify and proceed to the ticket counter. They inquire with the ticket clerk about the exit stop, the next mode of transport and the available tickets. The travelers travel as a family and prefer to take advantage of any available round trip and family discounts. The price of the tickets offered by the clerk exceeds the travelers' cash supply and they inquire about the form of money they could use. They pay with a credit card and buy the tickets. The cashier indicates that the tickets they have just purchased are ready to be used without validation. They keep their tickets.

Figure 2-9. Second mode of transport in scenario 2.

Subsequently, the travelers walk following the signs for the platform to the train with direction Graz. They find the platform depicted in Figure 2-9, they see that the train labeled Graz is boarding, and they enter. In the train, a ticket inspector asks to see their tickets as a proof for payment for being transported by train. They show their tickets and continue their ride. The travelers identify the stop for exiting by counting the stops after entering. They confirm and exit when they hear the loudspeaker announcement and see the signs for Bruck an der Mur. Figure 2-6 depicts Bruck an der Mur and its relative position to Vienna.

Decision point 3: Bruck an der Mur train station

The passengers are now at the platform of the train station Bruck an der Mur in Steirmark. In the planning phase, the travelers have identified that they should use the bus after the train. They have also identified the bus destination, exiting stop and time schedule. They have confirmed this information with the clerk at the Südbahnhof train station. At the train platform, there are no visible signs for the exit or for the buses and they are assisted by the stationary artifact of the steps and the dynamic artifact of the moving cluster of people to find their way. They go down the steps following the cluster of exiting passengers and see the entrance hall of the station. The travelers follow the signs at the entrance hall and walk to the bus stops outside the station.

Figure 2-10. Bus stops outside the Bruck an der Mur train station

In the planning phase, the travelers have identified the bus destination namely Aflenz Kurort. The three different bus stops are located outside the train station are shown in Figure 2-10. The travelers identify the right bus stop by comparing the line destination and individual stops listed on the bus schedule at each bus stop with their destination Aflenz Kurort. At the bus stop, they compare the bus' planned departure time with the current time and they understand that they have missed the bus. The train that brought them to Bruck an der Mur had a ten minute delay, and the bus has already left. In the planning phase, the travelers have been informed that the lift is the next mode of transport after the bus and that the last lift to the top departs in less than two hours.

The next bus to Aflenz Kurort does not leave for at least another two hours. The travelers can not wait for the next bus. Instead they look for a taxi as an alternative mode to the lift. They walk to the taxi which waits for customers outside the train station shown in Figure 2-11. The travelers communicate their destination point to the taxi driver and ask for the price of the ride. They board the taxi, and the taxi driver drives to the lift station of Aflenz Kurort. The travelers pay the fair with local currency at the end of the ride and exit.

Figure 2-11. Third mode of transport in scenario 2.

Decision point 4: Aflenz Kurort lift station

The travelers are at the lift station. In the planning phase the travelers have identified the lift destination. They confirm the lift destination and the exiting stop and inquire about the tickets. The travelers travel as a family and they prefer to take advantage of any available round trip and family discounts. The lift operator offers them the tickets and they pay with local currency. Buying the tickets from the operator also validates the ticket and the travelers are now legitimate travelers. Subsequently, they board the lift. This lift has intermediate stops. The travelers have confirmed with the operator that their exit stop is the final stop of the lift. Thus they stay on the lift until the end of its route and exit at the top. Figure 2-12 shows the lift route to the top of Bürgeralm.

Decision point 5: Bürgeralm

The travelers are at the end station of the lift on Bürgeralm. In the conception and planning phases they have identified the hotel Naturfreunde on Bürgeralm as the final destination of their trip. They acquire information from other tourists to find their way to the hotel. They walk towards the indicated direction and confirm the name posted on the hotel they approach. It is their hotel.

Figure 2-12. Fourth mode of transport in scenario 2.

2.4 Summary of operations in case scenarios

Table 2-1 provides a list of the operations that the traveler of the first scenario performs during his trip. The choice of an action in preference to another action as a response to the question indicates the presence of a decision problem (Edwards 1967). The traveler needs to perform operations in response to these questions. Thus, each operation is related to information needs. Aristotle introduced his "know and act" relationships as early as the 4th century BC in his Nicomachean Ethics (Aristotelis).

The operations take the traveler from one state to another. For example when the passenger validates his ticket he becomes a legitimate traveler as opposed to illegitimate. At each decision point for wayfinding, similar operations are performed. Table 2-1 distinguishes between four levels of operations. Operation level 1 involves the overall trip from the airport to the hotel. This qualifies as a wayfinding task. In operation level 2, the overall wayfinding task consists of a spatial and a business component. In operation level 3, the spatial wayfinding operations are further refined to operations relevant to finding a place or being in a transportation vehicle. In the same operation level, the business operations are refined to legal transactions and holding operations for the traveler's purse, wallet, pocket or bank account. In operation level 4, the spatial component involves operations related to finding the way to platforms, bus stops, vending machines, validation slots or identifying stops for exiting and entering. The business aspect involves operations related to business transactions such as identifying currency, buying and validating tickets etc.

Decision Point for Wayfinding Mode	Operation level 2	Operation level 3	Operation level 4
1: The airport			
	Take bus to destination		
		Find bus stop	
			Find way to bus stop
			Identify bus stop
		Be in bus	
			Identify upcoming bus
			Board bus
			Identify bus stop for exiting
			Exit bus
	Become legitimate passenger		
		Buy ticket	
			Identify currency
			Identify form of money
			Identify type of ticket
		Hold legitimate ticket	
			Identify need for validation
			Keep ticket
2: City Air Terminal, Hotel Hilton			
	Take metro to destination		
		Find metro	
			Find the entrance to the metro station
			Identify metro line
			Find way to the platform
			Wait
		Find vending machine	
			Find way to vending machine
		Find validation artifact	
			Find way to validation artifact
		Be in metro	
			Identify upcoming metro
			Board metro car
			Identify metro stop for exiting
			Exit metro
	Become legitimate passenger		
		Buy ticket	
			Identify currency
			Identify form of money
			Identify type of ticket
		Hold legitimate ticket	
			Identify need for validation
			Validate ticket
			Keep ticket
3: Karlsplatz			
	Go to hotel		
		Find Sacher hotel	
			Identify exit of metro station
			Find way to the Opera
			Find way to the street
			Find way to Sacher hotel

Table 2-1. Operation level 1: finding the way from the airport to the hotel on a Saturday with the use of public transport

The operation levels listed in Table 2-1 are related to the information needs of the different phases of the trip as presented in Section 2.2. Table 2-2 provides the correspondence between the different phases of the trip implementation and the operation levels and thus information needs. Operation level 1 corresponds to the conception phase and operation level 2 to the planning phase. Operation level 3 and all subsequent operation levels correspond to the execution phase. The structure of the table indicates that the trip is comprised of units of operations which are repeated throughout its execution. The table lists the following three types of units of operations situated in operation level 3:

1. spatial wayfinding unit: find vehicle stop, vending machine, and validation artifact;
2. "be in" unit: be in vehicle, hold legitimate ticket; and
3. business unit: purchase ticket.

The unit "be in" is referred to as container unit later in this thesis. Additional levels of operations include entering and exiting a vehicle, validating a ticket, etc. The concept of the units of operations is discussed in subsequent chapters of this thesis.

Operation level							**Trip phase**
1	*Wayfinding from airport to hotel*						Conception
2	*Take vehicle to destination*		*Become legitimate passenger*				Planning
3	*Find vehicle stop*	*Be in vehicle*	*Find vending machine*	*Find validation artifact*	*Purchase ticket*	*Hold legitimate ticket*	Execution

Table 2-2. Operation levels and trip implementation phases

The implementation of the second scenario as described in Section 2.3 is comprised of the same operations as the first scenario. A table presenting these operations would only add complexity without contributing any additional information and has thus been omitted. This observation leads to the consideration of a simpler case study for the modeling needs of this thesis.

2.5 Description of a reduced case

The complexity of the described scenarios leads us to regard a simpler case of using public transportation for the modeling needs. In the reduced case, the passenger moves from a street intersection in Vienna's 9th district to a street intersection in Vienna's neighboring 19th district. As the distance is far for walking, the traveler uses public transportation. The passenger needs answers to questions such as: Where is the vehicle stop? Which vehicle line? Which destination? What type of ticket? What form of payment? Is there a need for ticket validation? Where is the validation slot? Is this the right vehicle? Is this the stop for exiting?

2.5.1 The traveling environment

Figure 2-13 shows the street network in the area of Vienna considered in the reduced case. The area belongs mostly to a small part falling at the northern outskirts of Vienna's 9th District "Alsergrund" with the addition of the southern portion of Vienna's 19th District "Döbling" namely the area of "Oberdöbling". The thick dashed lines depict the routes of street car lines "37" and "38".

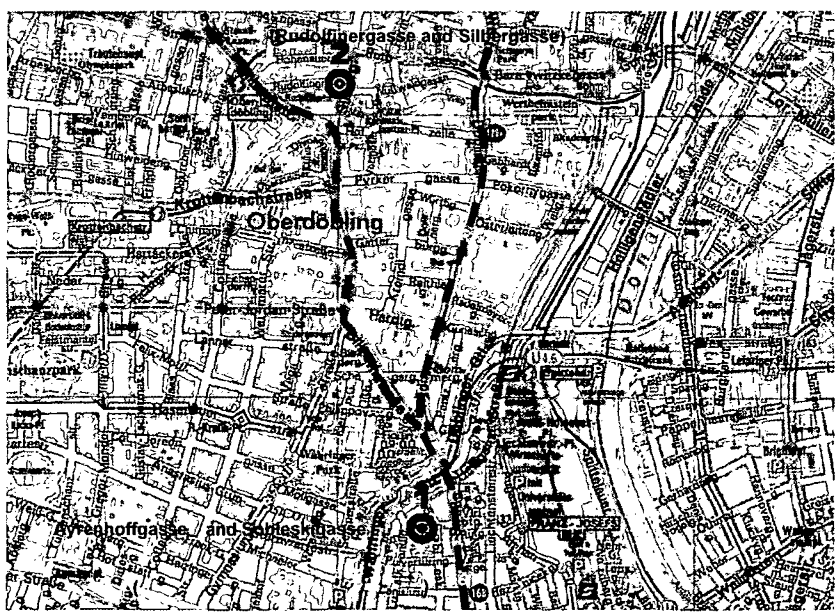


Figure 2-13. Vienna's northern portion of 9th and southern portion of the 19th districts

Both street car lines run on the same tracks for part of their routes and then follow their individual ways to their destinations. Figure 2-14 shows the maps of the street car lines "37" and "38". The encircled sections depict the focus areas of the reduced case. The information collected by the traveler in the conception and planning phases assist the traveler in the execution of the trip. The reduced case is subsequently used in the development of the conceptual and formal models of this thesis. The conceptual and formal models are described in Chapter 3 and Chapter 7 respectively.

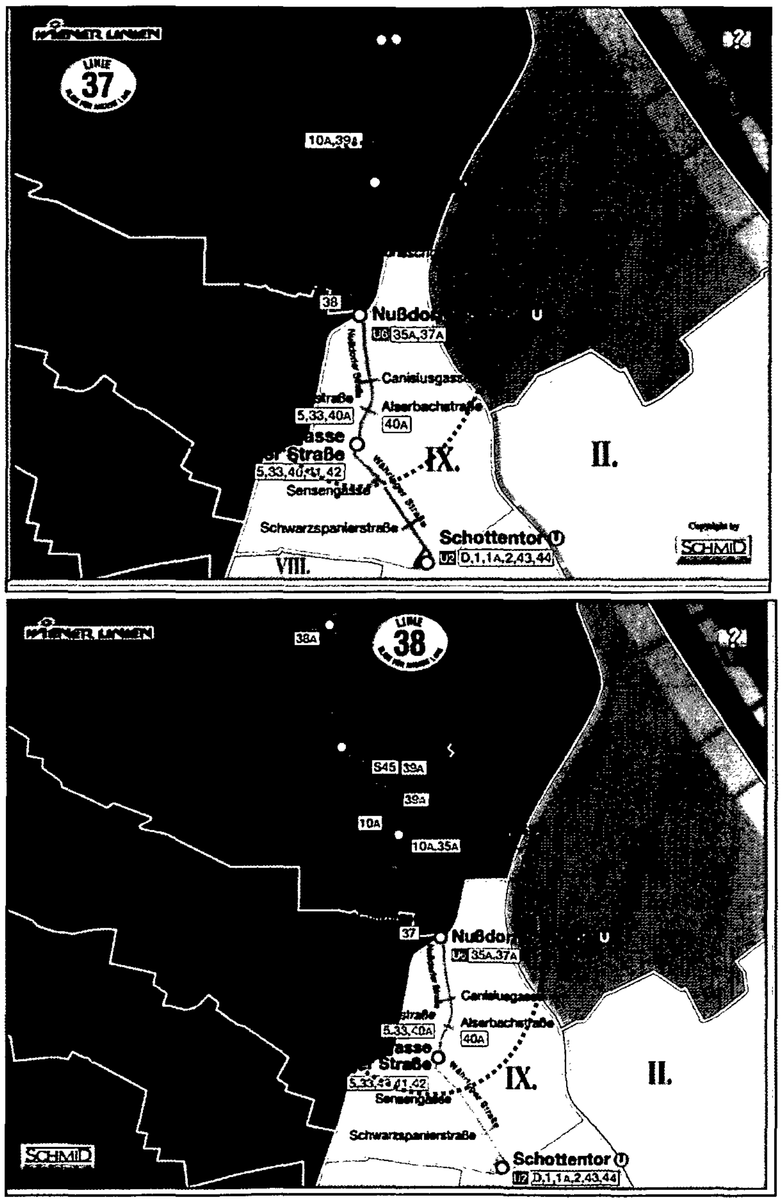


Figure 2-14. Map of street car lines 37 and 38. *(Source: Wiener Linien)*

2.5.2 Conception phase

In the reduced case the traveler is in the 9th District of Vienna near the street car stop Nussdorferstrasse and he goes to the 19th District of Vienna near the stop Oberdöbling. The intended time for the trip is a weekday and the mode of transport is public transportation. Figure 2-13 shows the beginning and end points of the traveler's trip.

2.5.3 Planning phase

By looking at the map, the traveler identifies that he should use the street car line 38 in the direction of Grinzing. His destination is within walking distance from the stop Oberdöbling. The need for payment is known to the traveler from experience and he has local currency.

2.5.4 Execution phase

The traveler uses the information collected during the conception and planning phases to assist him in the execution of his trip. The traveler finds his way to the street car stop by looking at the street signs and by perceiving public transportation related artifacts. He buys a single one-way ticket at the tobacco store next to the bus stop and he pays in cash with local currency. He sees a space indicative to validation on the ticket but he can not perceive any validation artifact in the vicinity. He waits at the stop Nussdorferstrasse and he selects the street car line 38 in the direction of Grinzing. He ignores the lines 37 and 38 in the direction of Schottentor. He boards the car and sees a validation artifact inside the vehicle. He validates his ticket, keeps it, and waits inside the moving vehicle until the stop Oberdöbling. There he exits. He finds his way to his final destination by looking at the street signs.

Decision Point for Wayfinding Mode	Operation level 2	Operation level 3	Operation level 4
1: Nussdorferstrasse			
	Take street car to destination		
		Find street car	
			Find way to the street car stop
			Identify street car line
			Wait
		Find ticket vending place	
			Find way to ticket vending place
		Find validation artifact	
			Find way to validation artifact
		Be in street car	
			Identify upcoming street car
			Board street car
			Find way to validation artifact
			Identify street car stop for exiting
			Exit street car
	Become legitimate passenger		
		Buy ticket	
			Identify currency
			Identify form of money
			Identify type of ticket
		Hold legitimate ticket	
			Identify need for validation
			Validate ticket
			Keep ticket
2: Oberdöbling			
	Go to final destination		
		Find street intersection	
			Find way to street intersection

Table 2-3. Operation level 1: finding the way from an intersection in the 9th District to an intersection in the 19th District with the use of public transport

Table 2-3 provides a list of the operations that the traveler of the reduced case performs during his trip. Each operation is connected to information needs. Similar to the previous scenarios, four levels of operations are identified and the overall wayfinding task is comprised of a *spatial* and a *business* component.

2.6 Summary

This chapter described the two motivating scenarios and the reduced case of wayfinding with the use of public transportation. The reduced case leads to the conceptual and formal models presented in subsequent chapters of this thesis. The descriptions show that the operations performed during a trip with public transportation and thus the related information are hierarchical refinements of coarse, medium and fine levels of detail. The overall wayfinding task found in the highest level is further refined into spatial and business components. Subsequent operation levels include operations such as buy, hold, board, etc.

The scenarios and the reduced case indicate that each trip is comprised of units of operations which are repeated. The description distinguishes between three phases of the trip implementation and presents the information needed in each phase. The scenarios demonstrate that there is a correspondence between the implementation phase of a trip and the operation level thus to the level of detail in the information needed. The information collected in the conception and planning phases of each trip, including information attributed to the traveler's previous experiences, assists him in the execution phase. This information is hereinafter in this thesis associated to the traveler's "mental plan" and is described in detail in Chapter 4.

3. SUPPORTING THEORIES

> *"...What characterizes experience is the influence of past occurrences on present reactions. When you offer a coin to an automatic machine, it reacts precisely as it has done in former occasions. It does not get to know that the offer of a coin means a desire for a ticket, or whatever it may be, and it reacts no more promptly than it did before. The man at ticket office, on the contrary, learns from experience to react more quickly and to less direct stimuli. That is what leads us to call him intelligent."*
>
> *Russell-Portraits from Memory: And Other Essays, p153*

This chapter describes the theories which support the conceptual and formal model. It provides a foundation for the subsequent Chapter 4 which describes the components of the conceptual model. This chapter examines multi-agent systems and cognitive agents to model the passenger's behavior as he perceives his environment and reacts upon it.

The space which is shared by the passenger and the transportation vehicle consist of discrete points. This space possesses affordances which are perceived and interpreted by the agent. The chapter discusses the concepts of *knowledge in the world* and the *affordances*. Signs, more specifically, transportation related signs are perceived and decoded by the agent. The theory of signs is also discussed in this chapter. The transportation vehicle is a container which locomotes through the transportation network following a schedule. Finally the chapter concludes with the discussion of the system epistemology and ontology.

3.1 Multi-agent architecture

The fable of the child who was fed and cleaned by one of a king's maids but was never spoken to is widely known. The king's desire was to investigate the development of the child's ability to speak and vocalize in such an environment. While the cruel king never had the opportunity to test this because the child died very young, the tale provides us with a great insight of distributed artificial intelligent. As Dreyfus says, "*...sub-worlds are not related like isolable physical systems to larger systems they compose; rather they are local elaborations of the whole they presuppose...*" (Dreyfus 1981). The fable and Dreyfus' quote suggest that systems are comprised of sub-systems influencing one another and show the concept of structures by interaction and the collective intelligence (Minsky 1985).

Artificial intelligence is the technology of information processing for reasoning, learning and perception (Tanimoto 1987). Bond and Gasser suggest that distributed artificial intelligent consists of areas of distributed problem solving and multi-agent systems (Bond and Gasser 1988). The first is associated with the top down approach of dividing a problem into sub-problems and pursuing solutions by individual agents which contribute to the solving of the initial problem. The second starts on the definition of the individual agents (Bittner 2001).

There is no agreed definition of intelligence. Typically intelligence is associated with problem solving. Tanimoto includes the following aspects in his definition of intelligence: intuition, common sense, judgment, creativity, goal directness, reasoning, knowledge, and beliefs (Tanimoto 1987). Distributed artificial intelligence focuses on systems or agents which work together towards solving a problem. This research considers agent-based modeling as the conceptual framework for the representation of the realm of wayfinding in public transportation. According to Ferber, a multi-agent system encompasses the elements listed in Table 3-1. The next section of this chapter discusses the concept of agents and focuses on the cognitive type of agents used to model a passenger who performs a wayfinding task.

An environment namely a space which has a volume. *A set of passive objects which posses a location in the space where they are found. An agent can perceive, create, destroy and modify the objects.* *A subset of the set of objects representing the active agents.* *A set of relations linking objects to each other.* *A set of operations which are used by the agents when acting upon the objects.* *The laws of the operations representing the application of these operations and the reaction of the world.*

Table 3-1 Elements of multi-agent systems from (Ferber 1998)

3.2 The cognitive agent

The cognitive agent in this research is a passenger who cognizes about himself and about his surroundings, namely his environment. The two components of this pair are described in this section starting with the definition of an agent and continuing with what it means by an agent that is cognitive. A number of attempts have been made in research to define an *agent*. A non-exhaustive selection of verbs associated with the definition of agent includes "believe", "intent", "desire", "must", "think", "learn", "perceive", "remember", "conclude", "decide", "act", and "react". Variations include the elements of "posses skills" and "self reproduce" (Ferber 1998). Russell's and Norvig's definition of an agent associates him with perception and action (Russell and Norvig 1995). According to Russell and Norvig an agent is anything which can perceive the environment and act upon it. This definition encompasses human beings, computer programs, and sensors.

A large number of classification schemes, mostly from the area of artificial intelligence are derived from the combination of various characteristics of agents. A limited classification based on the order of input and the agent's belief, obligation, intention, and desire yields 144 distinct partial orders of output (Dastani and van der Torre 2002).

The "*agent's architecture characterizes its internal structure, that is the principle of organization which subtends the arrangement of its various components*" (Ferber 1998). Table 3-2 provides a summarized list of features which are possessed by an agent.

He can act upon the environment.
He can communicate with other agents.
He is driven by objectives.
He possesses resources on his own.
He can perceive the environment.
He has a partial representation of his environment.
He possesses skills and can offer services.
He may be able to reproduce himself.
He tends to satisfy his objectives using his skills, resources, percepts, representations and output of his communication.

Table 3-2: Capabilities of the physical or virtual agent–based on (Ferber 1998)

The passenger-agent is goal driven and locomotes in a static, episodic and discrete environment. A *goal driven* agent is an agent whose decision making strategy depends upon the accomplishment of his goal. The *episodic* environment where the agent moves allows him to experience one episode of the world at a time and then move on to the next episode. He does not have to keep track of the history of the world or of his own actions. In this sense, the agent has no memory. A different kind of memory which is associated with previous experience and learning is present in the list of the agent's cognitive abilities. This is termed "a priori knowledge" for the sake of this research and it is considered unchanged during the trip. The agent possesses a priori knowledge before he embarks on his task. The a priori knowledge assists the agent to differentiate, for example, between a private car and a public transportation vehicle or to know that he needs to pay for the service of transport etc. In the scenarios and the reduced case described in Chapter

2, this knowledge is presented as being the result of the trip's preparation and the passenger's previous experience. In the reduced case, no change takes place in the environment while the passenger is on his trip. This is a *static* environment.

The agent's goal is the overall goal of the trip, namely to reach a specific destination with public transportation. This is a wayfinding process with a spatial and a business component, and it is situated at the uppermost operation level as discussed in Chapter 2. It is also possible to consider that the goal of reaching a destination is a synthesis of sub-goals, namely the goals of all individual episodes. This is the case in a multi modal trip where each mode qualifies as a separate wayfinding task with its own goal.

3.3 Knowledge in the world, image schemata and affordances

Norman states that there is information embedded in the world which counterbalances the need for learning. When finding the way through a city or operating a machine, people use this external information stored in the world (Norman 1988). The use of knowledge in the world reduces the knowledge that the people have to store in their heads in order to perform a task. This leads to a tradeoff between knowledge stored in the head and that stored in the world. This tradeoff determines what has to be learned. In the realm of a wayfinding task with public transportation, there is increased knowledge as in the street environment or in the ticket vending machines which we regard as part of the knowledge in the world. The knowledge in the world informs the agent, for example, that he can not exit though a closed door of a vehicle and that he should wait for the bus at the bus stop and not in the middle of the street where there is traffic. The knowledge in the world is related to affordances.

Objects carry by default *affordances* described by Gibson. For example, a chair affords sitting and a handle affords turning (Gibson 1986) and, as considered in this research a validation machine affords to have a ticket inserted to be validated. The agent perceives the affordances and uses the objects (Norman 1988). Gaver expands the notion of affordances to those which lead to additional affordances relevant to subsequent operations exercised upon objects (Gaver 1991). For example, ticket validation often involves inserting the ticket through a slot and subsequently passing through a rotating gate. This forms a sequential affordance of the validation gate. Figure 3-1 shows the affordances perceived at a ticket vending machine. The ticket vending machine has buttons for pushing and selecting among options, openings for receiving tickets or change, and slots for inserting coins, bank notes, and credit cards. These patterns and shapes relate to the notion of image schemata as described below.

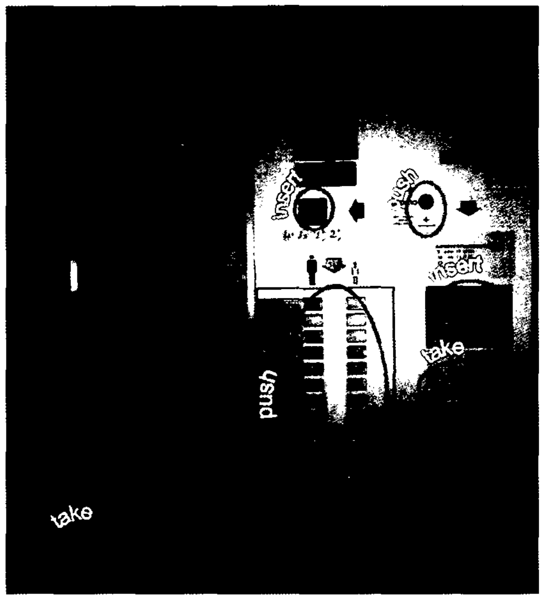


Figure 3-1. Affordances perceived at a ticket vending machine

Image schemata are recurrent patterns, shapes, and regularities of ordering activities used in the interpretation of the knowledge in the world (Johnson 1987). The image schemata constitute meaningful structures that include bodily movements, manipulation of objects, and perceptual interactions. The assemblage of elements which comprise a bus stop constitutes the image schema of a bus stop. The structure of the image schema of a bus stop is meaningful to a passenger. For this reason he is able to perceive and interpret a bus stop in diverse environments with similar structures, for example, in large European cities where typically includes a post holding a sign, a bench and markings on the street. The image schemata include the notion of force with its subsequent components of interaction, directionality, path of motion, origin, intensity and sequence. The image schema of a moving public transportation vehicle encompasses the components of path of motion, and force.

Lakoff defines the *container* as a bounded system which can hold a content and has a lower and upper bound. The vehicle is a bounded container which holds passengers, the driver, seats, etc. The passenger's pocket is also considered a container. A container has the properties of quantity, emptiness, misplacement, and containing elements that can be removed (Lakoff and Johnson 1980; Lakoff 1988). Lakoff's definition applies beyond the commonly known structures such as bottles, glasses, etc. Timpf considers the container as an aggregated hierarchical structure (Timpf and Frank 1997). Humans perceive image schemata and affordances and classify during perception. Typically, humans are able to distinguish and label a transportation vehicle as such. They are also able to distinguish and label operations.

3.4 The theory of signs

The conventional definitions refer to a sign as perceptively suggesting the existence of a fact, condition or quality which is not immediately evident or as encoding a communicative message (Morris 1973) (WordNet 2.0). Eco states that a sign is "*something that stands for something else in its absence*" (Eco 1997). As perception is linked to receptors addressing various parts of our sensory system, the signs too are linked to our different senses. The theory of signs is semiotics.

Pierce introduced the triple nature of a sign (Pierce 1992; Eco 1997; Priss 2001). He states that icons, indices, and symbols are the three principal natures of signs. He suggests that reasoning involves all three natures of signs, and he claims that it is closely related to perceiving and interpreting signs. Thus, he stresses the relationship between logic and semiotics (Pierce 1992). He refers to the *likeness* as the mapping of icons, sounds and gestures of pictures. *Indication (indices)* refers to the experience of perceiving similar signs in the past. The *symbol* connects the sign to the meaning and the usage. Symbols are *"applicable to whatever may be found to realize the idea connected to the world"*(Pierce 1992).

Dewey differentiates between *natural* and *intentional* signs. He refers to speech as an intentional sign and to clouds and footprints as natural signs. Pierce and Dewey spoke of the capacity of mind to pay attention to one element while ignoring others (Dewey 1938)

(Pierce 1992). Eco writes that *"It can be maintained that semiosic processes are involved in the recognition of the known, because it is precisely a matter of relating sense data to (conceptual and semantic) models..."* (Eco 1997).

The passenger is confronted with a world of signs not only the commonly understood street and transportation signs, but also signs in the sense described by Aristotle as the impression that an object leaves to a mind (Aristotle 1941 (reprint)). According to the attraction of attention described by Piece (Eco 1997), when a passenger approaches an intersection, the irregularity that he observes in the road pattern for example and the corner structures of the approaching buildings provide him with hints that he should look for an intersection and hence for the name of the intersecting street. Figure 3-2 displays such a situation where the passenger who locomotes in the direction from A to B may want to be attentive to the street-name signs as his field of vision, depicted here two-dimensionally, allows him to perceive the approaching intersection. According to Dewey and Pierce, a sign attracts the attention of a perceiver, thus causing him to ignore other elements in the environment. The passenger focuses his attention on what is important for his task and ignores the "perceptual noise", namely all other irrelevant elements. This keeps the passenger on track. While trying to find the ticket vending machine, the bus stop or the entrance to the subway station the signs as discussed here are used.

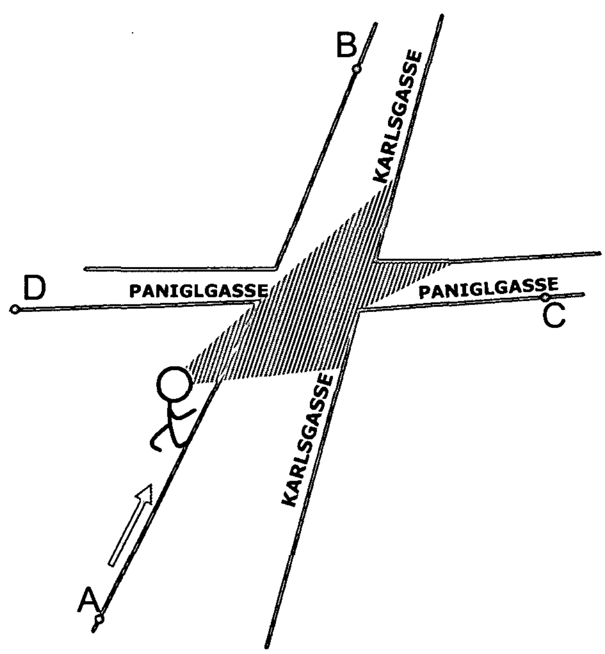


Figure 3-2. A passenger perceives an intersection

3.5 Spatial wayfinding and spatial decisions

The process of finding the way from an origin to a destination is defined as a wayfinding (Golledge 1999). The knowledge stored in the world, the affordances, the image schemata, and signs, discussed in Sections 3.3 and 3.4, assist the agent to navigate in the environment. Acquisition, mental representation, use, and communication of spatial knowledge contribute to people's spatial cognition (Mark, Freksa et al. 1999). A different type of assistance in wayfinding is based on spatial coding and spatial experience. Newcombe and Huttenlocher propose a comprehensive scheme of spatial coding associated to external landmarks and to one's self. In the first category, they differentiate between "*cue learning*" and "*place learning*". "Cue learning" provides an association between the physical object which is to be located and an external landmark. "Place learning" refers to coding of distances and directions in relation to external landmarks. In the category of one's self referenced spatial coding Newcombe and Huttenlocher distinguish between "*response learning*" and "*dead reckoning*". "Response learning", often mentioned as sensorimotor coding, is the codification of a location or of a path through an assemblage of body movements, and "dead reckoning" is the coding which takes place when a location is linked to distance and direction relevant to someone's current position and to subsequent movement (Newcombe and Huttenlocher 2000). Supplementary, the coding of a spatial experience is also attributed to stationary, moving, and dynamic local artifacts (Pontikakis 2005).

A *cognitive map* is a mental representation of the wayfinding environment (Golledge 2005) (Kuipers 1978; Kuipers 1982). Golledge proposes a set of questions asked through a cognitive map such as: "is it there?", "how can I go there?", "what is the shortest way?". Allen differentiates wayfinding in connection to the goal of the agent according to: wayfinding to a familiar destination, wayfinding to a familiar origin (walks for pleasure), and wayfinding to an unfamiliar destination (Allen 1997; Allen 1999). Raubal proposes a model for wayfinding in unfamiliar environment, in airports. He models the agent's knowledge in the head and the knowledge in the world and relates the wayfinding to spatial decision making (Raubal 2001). *Spatial decisions* depend on alternatives and yield a value for each alternative (Edwards 1967). People use qualitative and topological criteria for their spatial decisions such as relative position, similarity, proximity, enclosure, etc (Frank 1992) (Freksa 1992) (Piaget and Inhelder 1967). In this research, a spatial decision is regarded in reference to the operation it invokes. For example, the operation "walk" can be connected to the decision to walk "left" or "right" with results "getting lost" or "being on track" as follows:

walk (left) = getting lost

walk (right) = be on track

The passenger navigates in the public transportation space confronted with questions for which he gives answers and takes decisions based on his perception, his plan, and his knowledge as depicted in Figure 3-3.

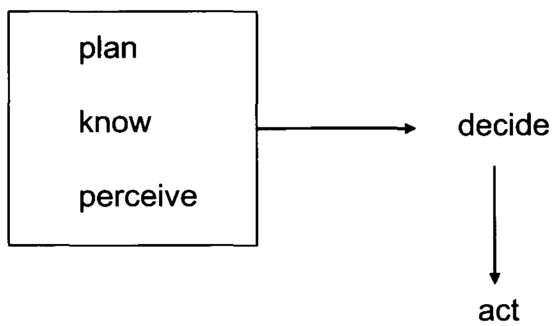


Figure 3-3. "Decide – act" scheme of an agent

3.6 Business process and public transportation

A service is an intangible economic activity that contributes directly to the satisfaction of the human desire (Pass, Lowes et al. 2000). Transportation is a service offered by a transportation provider to passengers. The service concept encompasses the following components (Kivel 2006): 1) External service value which refers to a defined service quality cost for the customer; 2) Customer satisfaction which includes attractive values and a customer needs oriented service; and 3) Customer loyalty in the sense of retention, repeat business, and referral and resulting in growth and profitability.

A business model explains payment in terms of: who, how much, and how often he pays. The *service profit chain* includes 1) the operating strategy and service delivery system and 2) the service concept which represent the result for the customer. In the case of public transportation the first refers to the transportation provider and the second to the passenger, the focus of this research. A business model assists us in understanding, describing and predicting the "*activity of buying and selling goods and services*" (Osterwalder 2004). Osterwalder states that a business model is an abstract representation of the business logic of a company. He suggests adopting a framework that consists of the following four areas:

a) the *product* which describes the products and the value proposition;

b) the *customer interface* which includes the target customer, distribution channel, and the relationship with the customer;

c) the *infrastructure management* which explains value configuration based on arrangement of activities and resources, the capability to execute repeatable patterns, and the partnership between two or more companies; and

d) the *financial aspects* which describes the cost structure, and revenue model (Osterwalder 2004).

Kivel defines the business model for public transportation as the method by which public transportation uses its resources namely, cash, technology and people to provide its customers with better value in comparison to its competitors, and to make a profit in doing so (Kivel 2006). Osterwalder suggests that the term "conceptual business model" is often confused with "business modeling" and "business process" (Osterwalder 2004). The conceptual business model is an abstract description of a simplified representation of a particular business. The business modeling is a process related to a business model. The business model describes the business process.

3.6.1 Value in a business model

This research focuses on a passenger's wayfinding with public transportation. In Osterwalder's suggested framework, the main objective of most components is to create value for the customer (Osterwalder 2004). *Value* is defined based on a multitude of factors. They vary from the scarcity of a product or service, the labor required for production (Smith 1986 (original 1776)), the consumer preference and judgment of utility

(Woo 1992), the combination of benefit and price, and the added value for the customer (Krek 2002).

The value proposition in a business model represents "*an overall view of the firm's bundles of products and services that together represent value for a specific customer segment*" and includes a set of *elementary offerings* (Osterwalder 2004). An elementary offering is characterized by its description, reasoning, life cycle, and price level. *Price discrimination* refers to the combination of the following: a) selling units of a product or service to b) different or the same customer segment, for c) the same or different prices. Value proposition and value discrimination in ticketing are part of the business process of the public transportation business model.

3.6.2 Business modeling in public transportation

Funding for public transportation typically comes from a combination of government subsidy and fare collection. Reasons for improving and expanding public transportation do not only include the transported passenger and the business itself, but also the impact that public transportation has in reducing air pollution and traffic congestion. This research focuses on the transported passenger. The business aspect that is visible to the passenger is related to ticketing. Like money, and property titles, tickets are part of the institutional reality (Searle 1995). Tickets have a value which is a function of their *price* and of the *penalty* that the user has to pay in the case of absence or inappropriate use of a ticket. For this reason, the author of this research terms the ticket as a "*proof of payment for transport*".

Ticketing is an example of the passenger's difficulty in dealing with the value proposition and price discrimination embedded in a business model of public transportation. Table 3-3 lists the components for known elementary offerings in ticketing. Each elementary ticket offering includes the components of type of unit, customer segment, life cycle, and reasoning in combination with the resulting price. The price can also depend on the payment method and the deposited amount.

		Description	
Price	Units	Time coved Trips covered Modes allowed Area covered Passengers traveling	Payment method: cash, credit
	Customer	Age: senior citizen, child Education: student Financial state: unemployed, etc. Social state: refugee, handicapped, etc.	
	Life Cycle	Purchasing: in advanced, during ride, etc. Use: electronic (SMS, Etix, etc), paper form, tokens, etc Validation: validated at the time of purchasing, through slots, magnetic strip card, etc. Transfer among others	Deposited amount
	Reasoning	Benefits of public transportation for the customer (transport of passenger in timely, affordable, and environmentally friendly way) Reduced risks Reduced effort and convenience	

Table 3-3. Price determination in elementary offerings of tickets in public transportation

In several cities authorities provide free rides at certain areas or at certain times to alleviate traffic congestion. In this case, passengers pay the cost mostly through taxes. An additional incentive for free rides is to motivate citizens to perform physical movement and thus improve health conditions. Typically however, ticketing is a vital part of the business aspect of public transportation.

3.7 Modeling situations in public transportation

There two types of models in public transportation, those that are stochastic and those that are based on stochastic elements. Graham et al proposes the use of mathematical modeling in the domain of operation research for public transportation systems that includes relations, decisions, constraints, and performance measures(Graham, Cassady et al. 2000). In their modeling, they propose a set of classes of objects and operations that include: cargo, passenger, movement, vehicle, route, terminal, infrastructure, material handling equipment, passenger handling equipment, personnel, miscellaneous equipment, procedures and policies, disruptions.

DiFebbraro et al model intermodal passenger transportation systems with nodes, macronodes, links, inner links, and events. A station is intermodal if it facilitates more than one mode of transportation. A station that serves a mode of transport is regarded as a node. Macronode is a combination of nodes, link is the path of a mode of transport, and inner link is the path traveled by the passenger to transfer between two nodes of a macronode. Event is anything that can cause a change in the system and can be normal or stochastic. The arrivals and departures of vehicles constitute normal events while traffic jams and breakdown vehicles are stochastic events (DiFebbraro, Recagno et al. 1996).

Lott et al simulates an intermodal train station. He presents passenger interaction areas such as: waiting areas, corridor, entry and exit areas, open space, toilets, ticketing areas, platforms, etc. The model provides information on passenger activity such as boarding and exiting, occupancy and segment time for the passenger. The segment time includes the flow of passengers, size and time of waiting lines, demand for services, etc. (Lott 1993).

3.8 Epistemology

Epistemology expresses the reasoning and expression of knowledge of a domain or of a theme. The word originates from the Greek and consists of two components. The first component "epistamai" means "to know" while the second, "logos", has a dual nature, namely reasoning and speech. The latter manifests the connection between uttered speech and thinking. It was expressed by Plato in his dialogs (Plato 370 B.C. (reprint 1953)). Whorf, two and a half millenniums later, stated that our language drives our thinking (Whorf 1956). Chomsky reinstated and formalized the linkage (Chomsky 1988) (Chomsky 1967).

Section 2.4 introduced the units of operations as part of a trip's structure namely, the spatial wayfinding, the "be in", and the business unit. Each unit of operations involved in public transportation is characterized by an epistemological and a usage component. The epistemological component of each unit of operations expresses the knowledge which an agent utilizes either by retrieval of older stored knowledge or by acquisition of new information. The stored knowledge retrieved by the agent during the execution of the operations is referred to in this research as *a priori* knowledge and was introduced in

Section 3.2. The stored knowledge often represents years of accumulated experience. The second facet of the epistemological component pertains to the goal of each unit of operation and constitutes the teleological aspect which refers to the purpose or end of the operation involved (Britannica 2003).

3.9 The 5-tier ontology concept

Our effort to understand the world is linked to our ability to structure and decompose small pieces at a time. Ontology based analysis provides a significant tool for this effort and facilitates the passage from the conceptual to the formal stage of this research (Gruber 1993; Guarino 1995; Uschold and Grüninger 1996; Kuhn 2000). The 5-tier ontology concept was introduced by Frank (Frank 2001). It assists in decomposing and understanding pieces of the world at a time and hence constitutes a tool for formalization. It structures five tiers namely the human-independent reality found in tier 0; the observation and physical world in tier 1; the objects with properties in tier 2; the social reality in tier 3; and the subjective knowledge in tier 4. The above scheme was further developed (Frank 2001; Frank 2005-Draft) with tiers 3 and 4 placed in the same level.

In ACTOR 2002 (Frank 2003 Unpublished; Frank 2005-Draft) and in Ontology for GIS (Frank 2003 Unpublished; Frank 2005-Draft), Frank proposes an operation and process-based ontology as opposed to object-based. We use the activity in the observation level as the decomposition element in ontological analysis.

Tier 0 represents the physical reality which is common to all and it is placed in the realm of space and time. This belief of the external physical world which is independent of the observing subject is the basis of all natural science (Einstein 1934). In ontological studies tier 0 is referred to as the "physical world".

Tier 1 is the observation. It captures the output of observation by the observers namely the value of the observation and the observed parameters. Heisenberg states that the observer is part of the observation (Heisenberg 1983). A value can be obtained by a machine sensor or by a human or an animal. Machine measurements come close to a subject independent observation of reality(Frank 2005-Draft). Conventions are part of both machine and non-machine observations. The values of observations on human agents are connected to the manifestation of an agent's decisions and emotions in the space and time realm (Rottenbacher, Achatschitz et al. 2005, In preparation). Change is captured as change in value. The observed parameters are unique in space and time; therefore, two observations can not possibly be the same although they may be represented by the same value.

Tier 2 represents the objects, the operations, and aggregated or generalized instances. Tier 2 is where most of this research is situated. Categorization and instantiation is performed differently by different individuals and it is an integral part of the scope of every research. It is applied to the objects and to the operations upon and from these objects. Human agents perform a limited set of interactions with the environment including perceiving, moving, holding etc. (Frank 2005-Draft). These interactions are used in structuring ontological operations and algebras such as: "walk", "wait", "turn", "exit", and "do nothing".

Tier 3 is the subject, namely the cognitive agents and organizations. They are embedded in the environment which they perceive through sensors (Ferber 1998; Frank 2005-Draft). They accumulate up to date knowledge which they use to interpret messages and make decisions for achieving their goals.

Tier 4 is the cultural and institutional realm with the laws that govern every activity. Institutions, laws, language, payments are examples of this tier. Object identifiers are results of cultural and social process. The elements of this tier evolve slowly (Frank 2005-Draft). They form conventions and they are prone to subjective judgment. Under this scope, conventions glide between tier 1 - observation and tier 4 - cultural and institutional reality.

In Frank's later work, tiers 3 and 4 are part of a broader tier which encompasses what he calls the "information realm" (Frank 2005-Draft). Tiers 1, 2, and the information realm of tiers 3 and 4 are always in a context of space and time with varied frequencies and starting points. Figure 3-4 depicts a chain of processes. An operation which leads to a goal can be partitioned in sub-operations with sub-goals as shown in the case of the operation that leads to "Goal 3". The goal of a process can be the beginning of the next one. This figure depicts the hierarchies found in processes and operations. The process which leads to goal Goal1b is at a finer level than the process that leads to Goal2. Chapter 2 discussed the operation levels in wayfinding with public transportation.

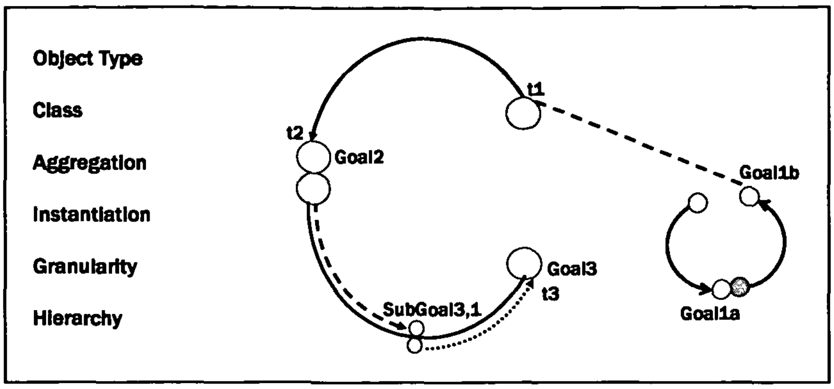


Figure 3-4. Ontological process chain

3.10 Summary

This chapter described the theories which are used in the formation of the conceptual and formal model. Initially the multi-agent systems and the cognitive agent theory were discussed to explain the passenger's behavior as he perceives his environment and reacts upon it. The knowledge in the world, the affordances, the image schemata, the signs, and the agent's spatial coding assist him to navigate through space. The chapter discusses the theory of the spatial wayfinding and decision and presents the business aspects of public transportation which are relevant for the passenger as described in the literature.

The novelty of this research is that it looks at the wayfinding in public transportation while integrating the business aspect for the passenger. The chapter also discusses the topic of epistemology as the reasoning about the knowledge and the goal of the operation. Finally it introduces the 5-tier ontology concept which is used for describing the operation in subsequent chapters of this thesis.

4. CONCEPTUAL MODEL

"These fundamental concepts and postulates, which cannot be farther reduced logically, form the essential part of a theory, which reason cannot touch. It is the grand object of all theory to make these irreducible elements as simple and as few in number as possible, without having to renounce the adequate representation of any empirical content whatever."

Einstein-Essays in Science, p15

The second chapter described the two motivating scenarios and the reduced case of this research and demonstrated the connection between the different phases of the trip implementation and the level of detail in the information needs. The conceptual model is built based on the simplified concrete reduced case. This chapter provides a detailed description of the conceptual model.

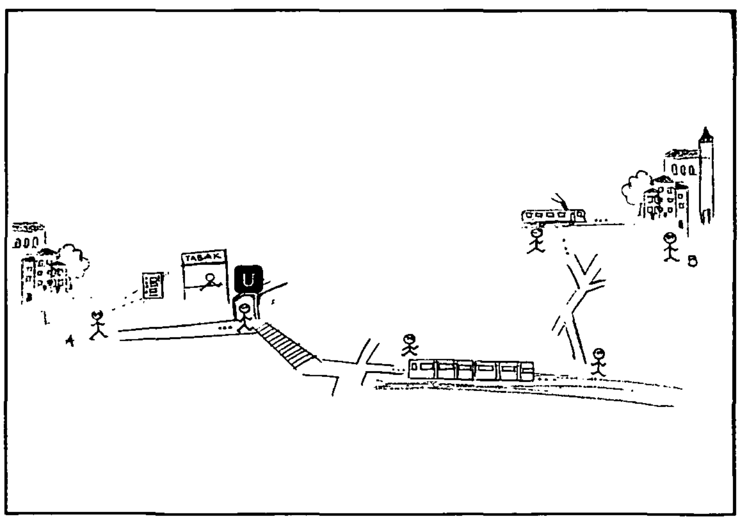


Figure 4-1. Overview of abstract case

4.1 Introduction

Figure 4-1 shows an abstract case of a passenger moving with public transportation. This case involves two modes of transport. The first mode is the metro while the second mode is the street car. The sketch in Figure 4-1 presents a typical situation. The residents of today's cities are confronted with such settings on a daily basis. When the passenger leaves point "A" to visit point "B", he ventures into a number of decisions and tasks. We relate these tasks to the operations described in Chapter 2. Chapter 2 introduced the different operation levels and it connected them with the level of detail in the information needs for their execution. The conceptual model facilitates the transition between the reduced case and the formal representation. The conceptual scheme of this research is comprised of the operations, the passenger, the ticket, the point network, and one or more vehicles.

4.2 Processes and operations in wayfinding with the use of public transport

A hierarchical structure of the operations involved in a wayfinding task with the use of public transportation emerged through the scenarios and reduced case presented in Chapter 2. Table 2-1 and Table 2-3 of Chapter 2 present this structure. Figure 4-2 presents a schematic of this structure. The schematic is not exhaustive and should be considered as a limited example. Additional operations not listed in Figure 4-2 are discussed throughout this thesis.

The top level task of going from one place to another with public transportation constitutes a wayfinding process. This is then decomposed into a spatial wayfinding and a business process. Moving down the hierarchical ladder, there are operations of finer detail such as "board", "validate", etc. These operations are related to axioms and algebras discussed later in this thesis.

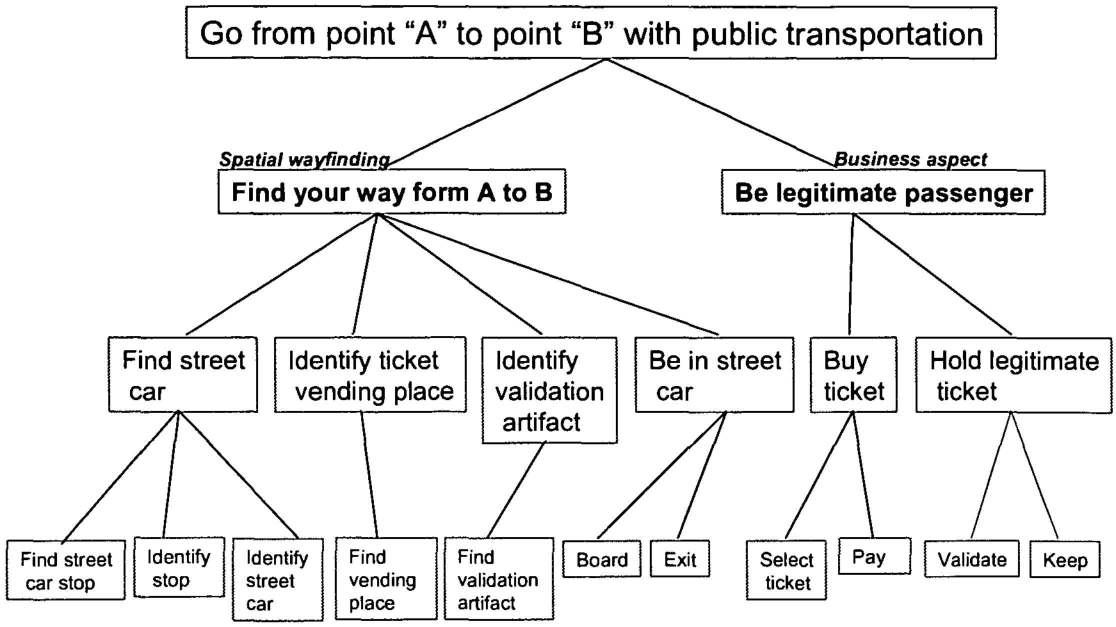


Figure 4-2 Structure of operations in wayfinding with public transportation

At decision points the passenger faces questions such as "Is this my vehicle stop?" These questions indicate a decision problem (Edwards 1967). Table 2-1 and Table 2-3 list operations performed by the passenger during the trip. At a vehicle stop, for example, the passenger faces the question "Is there a need for validating the ticket?" The passenger needs to decide whether to validate his ticket or not. The information needed for answering this question is found on the ticket and in the environment. The passenger perceives this information through signs and affordances. Perception, and identification of signs (Dewey 1938) and local artifacts (Pontikakis 2005) is directly relevant to a person's experience. If the passenger concludes that his ticket needs validation he performs the operation "Validate". The sequence of operations for the ticket validation at a fine level is listed below:

1. Perceive environment to identify need for validation.
2. Perceive environment to find validation artifact.
3. Perceive environment to validate ticket.

At a medium level, this sequence becomes:

1. Perceive environment.

2. Validate ticket.

At coarse level, it becomes:

1. Validate ticket.

Buying and validating a ticket, for example, are operations which often trigger the image of a passenger in someone's mind. However, similar operations can be attributed to someone who would like to go to the cinema, or to an amusement park, etc. The principles of categorization can be applied beyond the realm of objects into operations (Mark, Smith et al. 1999). The above examples of ticketing represent different instances of the same class of operations. This topic is further discussed in Chapter 7 of this thesis.

4.3 The passenger

The conceptual model encompasses an agent-based approach. The passenger acts as a cognitive agent who perceives, plans, and makes mental representations of the world, decides and acts upon his decisions, has a sum of money, and possesses or does not possess a ticket (Ferber 1998). The agent theory was discussed in Chapter 3.

4.3.1 Passenger's capabilities

The passenger does not bear any financial or time constraints. He has average perception capabilities through which he is able to respond to the affordances within the environment that are relevant to his trip. As a result he is, for example, able to walk on the sidewalk, to insert a coin in a slot in order to buy a ticket, to perceive the content of a bus sign, or to ascend and to descend a few steps in order to enter or exit a vehicle (Gibson 1986). The above are affordances of the sidewalk, the coin slot, the bus sign, and the steps which the passenger perceives.

4.3.2 The passenger's mental plan

The passenger typically conceives and plans his trip in advance. Chapter 2 showed that information is collected during the conception and planning phases which is later used in the execution phase of the trip. This information about the passenger's intended trip is either provided by another person in the form of instructions or is generated from sources such as a city map or a time table. It is typically combined with the passenger's previous experience, such as the need to pay for transportation. The collected information is associated with operations as shown in the previous section. The operations which refer to the conception and planning phase include the identification of the following elements:

1. *Beginning and ending of the trip.*
2. *Intended time of the trip.*
3. *Time schedule of intended vehicle.*
4. *Spatial wayfinding to vehicle stop.*
5. *Transport mode.*
6. *Need for payment.*
7. *Destination of the vehicle.*
8. *Decision point for change in wayfinding mode.*
9. *Spatial wayfinding to final destination or next transportation mode.*

Chapter 2 demonstrates that elements related to scheduling are used only if time is important. This depends on 1) objective factors such as the frequency of a transportation mode and the duration of the trip, 2) subjective factors such as importance of subsequent activities and impediments due to external conditions such as the weather, and 3) emotional state. The need for payment listed in the above list of elements is typically derived form experience.

The agent embarks on his trip having in mind a mental plan of his intended journey. The mental plan is a sequence of entries that are stored in the agent's mind (Denis 1997). The agent uses the mental plan to verify his current state and to move on to the next. An entry in the agent's mental plan relates to the spatial or to the business wayfinding of the trip. For example, it instructs him to move in a certain direction during his locomotion, to look for signs relevant to his task, to pay for the service of public transportation, and select transportation mode, line destination and boarding and exiting stops.

Mental plan entry Level 1	Mental plan entry Level 2	Mental plan entry Levels 3 and 4	Plan/ Experience
"at Ayrenhoffgasse and Sobieskigasse, start"			Plan
	"at Nussdorferstrasse street car stop, use street car 38 in the direction of Grinzing"		Plan
		"at Ayrenhoffgasse and Sobieskigasse, walk on Ayrenhoffgasse to Ayrenhoffgasse and Nussdorferstrasse"	Plan
	Wayfinding	"at Ayrenhoffgasse and Nussdorferstrasse, perceive street car stop"	Plan
		"at Ayrenhoffgasse and Nussdorferstrasse, walk to street car stop"	Plan
		"at Nussdorferstrasse street car stop, verify bus stop"	Experience
		"at Nussdorferstrasse street car stop, perceive street car 38 in the direction of Grinzing"	Plan
		"at Nussdorferstrasse street car stop, board street car 38 in the direction of Grinzing"	Plan
		"at Nussdorferstrasse street car stop, count 5 stops for getting off"	Plan
	Container	"at Oberdöbling street car stop, verify stop for getting off"	Experience
		"at Oberdöbling street car stop, get off"	Plan
	"at Nussdorferstrasse street car stop, become legitimate traveler"		Experience
		"at Nussdorferstrasse street car stop, perceive tobacco shop"	Experience
		"at Nussdorferstrasse street car stop, buy ticket"	Experience
	Business	"at Nussdorferstrasse street car stop, perceive validation artifact"	Experience
		"at Nussdorferstrasse street car stop, validate ticket"	Experience
	"at Oberdöbling street car stop, find Rudolfinergasse and Silbergasse"		Plan
	Wayfinding	"at Oberdöbling street car stop, perceive Rudolfinergasse and Silbergasse"	Plan
		"at Oberdöbling street car stop, walk on Rudolfinergasse to Rudolfinergasse and Silbergasse"	Plan
"at Rudolfinergasse and Silbergasse, stop"			Plan

Table 4-1. Mental plan in reduced case.

The mental plan for the reduced case is provided in Table 4-1. Each entry of the mental plan is a pair where the first element is a point in the street network, underlined text in Table 4-1, and the second element is the intended operation at this point. This is the structure of an instruction as suggested in the literature. Tversky and Denis propose that information structures in the form of depictions and descriptions, hold common components namely points, orientation and actions (Denis, Pazzaglia et al. 1999; Freksa 1999; Tversky and Lee 1999; Frank, Bittner et al. 2001). In the agent's mental plan the orientation element is the next entry's point. In this research, information structures of states and operations are also used as discussed in subsequent chapters.

The table distinguishes between planned information and information derived from experience. The information is in three levels of detail. The first level corresponds to a coarse level which includes only the starting and ending points. The second level introduces the two components of the trip namely the spatial and business component and the third provides detailed information on the operations involved in the spatial and the business component. Ultimately, a mental plan constitutes the passenger's vision of his trip stored as a set of location points and operations associated to these points. There are cases where a mental plan for the entire trip is partially generated at the beginning and is completed as the trip progresses. In these cases, the trip could be regarded as being comprised of smaller components. Passengers make mental plans in the sense described above on a daily basis. The passenger's mental plan is not necessary a correct plan.

4.3.3 Sum of money

The passenger has a sum of money. He can either carry it with him in a pocket, a wallet or a purse or he can store the money in a bank account. In either case, there is a container. Money can go into the pocket-container or bank-account-container and money can be removed. In both cases, there is an initial value of the content of the container. In this case, the container has a lower bound and indefinite upper bound. The passenger compares the value of his container with the ticket price when purchasing the ticket.

4.4 Ticketing

The passenger possesses some form of money and may have a proof of payment for transport, namely a ticket. The proof of payment allows the passenger to qualify as a legitimate user for the services rendered within the institutional reality of the public transportation system (Searle 1995), (Frank 2001). This is the business process of ticketing for a trip. In the reduced case it is introduced to the passenger in operation level 1 (Table 2-3).

4.4.1 Ticket purchasing

Figure 4-3 is a picture of an older version of a vending machine in Vienna, Austria. The complexity of the options offered through the vending interface is conveyed by the number of buttons, boxes, colors, symbols and, for a non-German speaking tourist, by the German language (Norman 1988). Figure 4-3 displays the type of information embedded in the ticket vending machine which is relevant for the traveler.

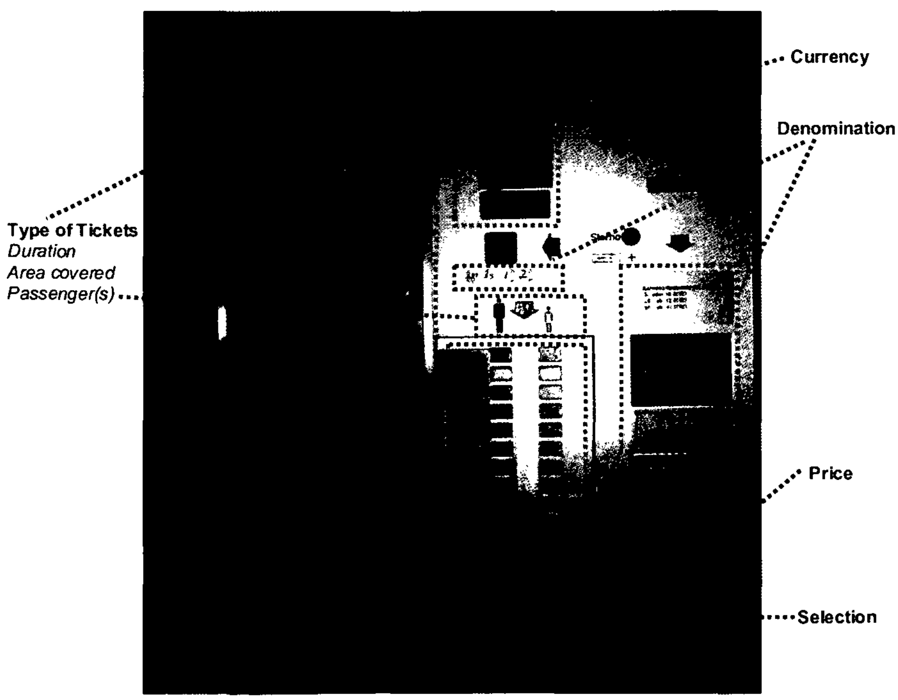


Figure 4-3. Older vending machine in Vienna still in use

This information is provided in the form of depictions or descriptions namely signs or text respectively and indicates the following:

1. ticket types based on trip duration, area covered during trip, type and number of passengers, number of trips, deposited amount, and types of modes
2. pricing,
3. currency, and
4. denomination.

This information is relevant to the operations described in Section 4.2 and constitutes only the purchasing aspect of the business model. The vending machine depicted in Figure 4-3 is still widely used and is often side by side with the new touch-screen variation. In the reduced case, the passenger buys a single one way ticket of full price (default option) with local currency at the tobacco shop next to the street car stop. The purchasing of the ticket is introduced in operation level 2 described in Table 2-3 and consists of a "container" and a 'business" operation. These operations are discussed further in Chapter 6. In the conceptual model the purchasing location is the street car stop.

4.4.2 Ticket validation

In some cases, when a passenger buys a ticket he can use it without validation, typically immediately after purchasing. This is conveyed to the passenger during purchasing verbally or with text and signs on the ticket. In these cases, the passenger needs to keep his ticket to legitimately ride the transportation vehicle. This is not always the case however. Having a ticket does not mean that the passenger is legitimate. The following cases of illegitimate passengers relevant to ticketing occur:

1. a passenger without a ticket,
2. a passenger who has the wrong ticker,
3. a passenger who has an appropriate but not validated ticket, and
4. a passenger who has a validated but inappropriate ticket.

The need for validation is indicated to the user in the depictions on the ticket itself, descriptions on the ticket or on the vending machine and through artifacts found in the environment. In the reduced case, the passenger validates his ticket after boarding the street car. For the conceptual model the location of validation is at the boarding station. . Ticket validation is introduced to the passenger in operation level 2 described in Table 2-3. Figure 4-4 shows the ticket validation gates on the way to the platform in a metro station in Vienna, Austria.

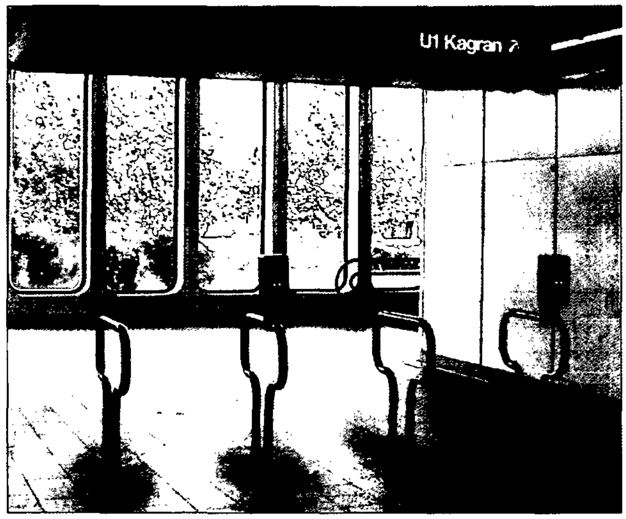


Figure 4-4. Ticket validation in a metro station in Vienna, Austria

4.5 The vehicle

Figure 4-5 shows a typical street car in Vienna. The vehicle depicted is of the line 37 in the direction of Hohe Warte, one of the lines considered in the reduced case. The vehicle is either in motion or at a stop. In the conceptual model, each street car is characterized by the line that it serves, by a direction, and by a sequence of stops that it passes. The line and destination of the intended vehicle should match the entry in the agent's mental plan, second level of entry shown in Table 4-1. The sequence of stops represents the predetermined route of a vehicle. In the reduced case the passenger uses line 38 in the direction of Grinzing.

4.5.1 The vehicle-container

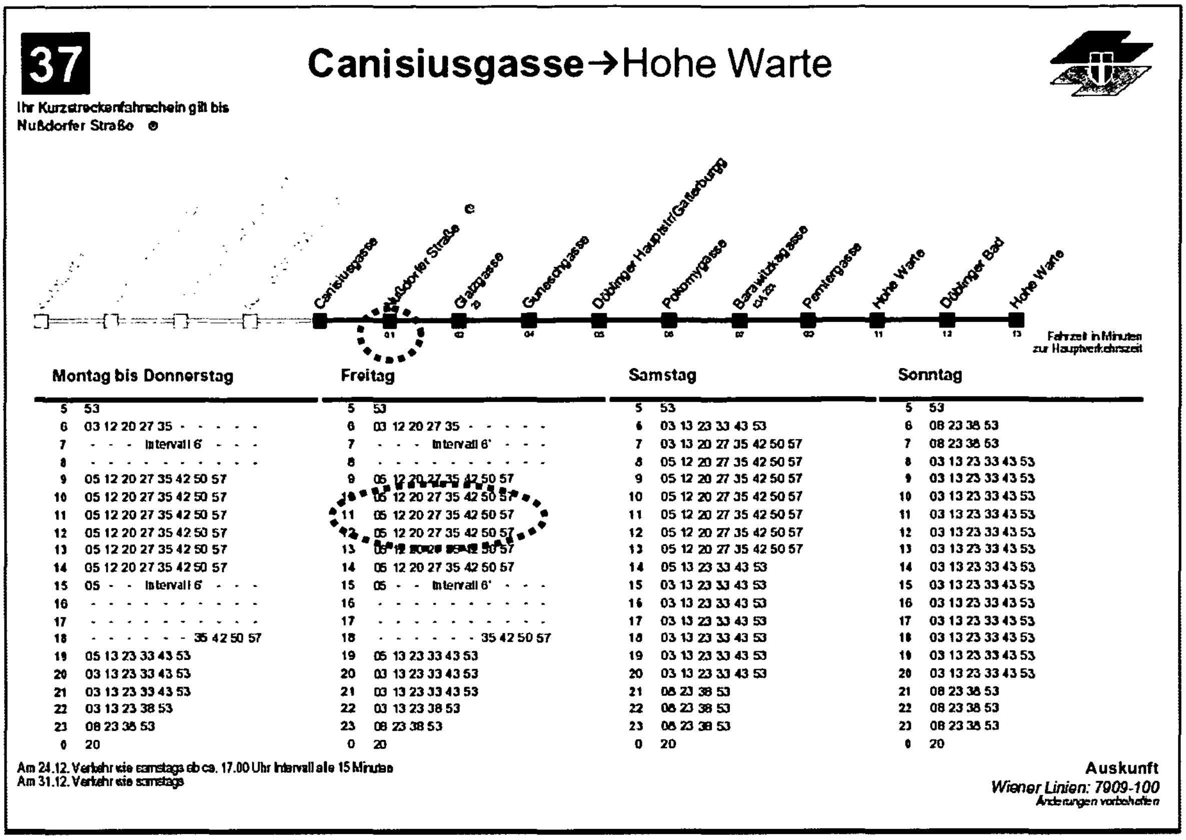
The vehicle in this research is a container. A passenger can move into the vehicle-container or he can move out of it. A vehicle-container can be empty with no passengers, full, or carrying some passengers and having room for additional passengers. This is relevant to the lower and upper bound of the container. The vehicle container has a lower and upper bound corresponding to being empty and full respectively. In this conceptual model there is always room in the vehicle-container for the passenger to enter, thus the container has no upper bound. Entering and exiting the vehicle are "container" operations. The container operations are discussed later in this thesis.

4.5.2 Temporal aspect of wayfinding with public transport- scheduling

The passenger does not influence the schedule of the vehicle. The street car scheduling introduces the aspect of the transport provider into the model. The passenger has no time constraints thus the vehicle schedule constitutes the only temporal aspect of conceptual model.

Figure 4-5. Street car in Vienna

The vehicle schedule is conceptualized as "scheduled time-delays". In the reduced case, there are two types of delays related to street cars. The first type of delay represents the time that a street car needs to reach a street car stop from its terminal. The second type of delay represents the elapsed time between two consecutive street cars leaving the terminal. The first type provides an association between each street car stop point and any street car. The conceptual model considers that the vehicles travel with the same speed, thus if a vehicle starts after another it will never overtake the first one. In the case of street cars, it is unfeasible for street cars to overtake one another.



37 Canisiusgasse→Hohe Warte

Ihr Kurzstreckenfahrschein gilt bis Nußdorfer Straße ⊕

Canisiusgasse 01 Nußdorfer Straße 02 Glatzgasse 04 Gunoldstgasse 05 Döblinger Hauptstr/Gatterburgg. 06 Pokornygasse 07 Barawitzkagasse 09 Pantzergasse 11 Hohe Warte 12 Döblinger Bad 13 Hohe Warte

Fahrzeit in Minuten zur Hauptverkehrszeit

	Montag bis Donnerstag		Freitag		Samstag		Sonntag
5	53	5	53	5	53	5	53
6	03 12 20 27 35 - - - - -	6	03 12 20 27 35 - - - - -	6	03 13 23 33 43 53	6	08 23 38 53
7	- - - Intervall 6' - - -	7	- - - Intervall 6' - - -	7	03 13 20 27 35 42 50 57	7	08 23 38 53
8	- - - - - - - - - - -	8	- - - - - - - - - - -	8	05 12 20 27 35 42 50 57	8	03 13 23 33 43 53
9	05 12 20 27 35 42 50 57	9	05 12 20 27 35 42 50 57	9	05 12 20 27 35 42 50 57	9	03 13 23 33 43 53
10	05 12 20 27 35 42 50 57	10	05 12 20 27 35 42 50 57	10	05 12 20 27 35 42 50 57	10	03 13 23 33 43 53
11	05 12 20 27 35 42 50 57	11	05 12 20 27 35 42 50 57	11	05 12 20 27 35 42 50 57	11	03 13 23 33 43 53
12	05 12 20 27 35 42 50 57	12	05 12 20 27 35 42 50 57	12	05 12 20 27 35 42 50 57	12	03 13 23 33 43 53
13	05 12 20 27 35 42 50 57	13	05 12 [illegible] 50 57	13	05 12 20 27 35 42 50 57	13	03 13 23 33 43 53
14	05 12 20 27 35 42 50 57	14	05 12 20 27 35 42 50 57	14	05 13 23 33 43 53	14	03 13 23 33 43 53
15	05 - - Intervall 6' - - -	15	05 - - Intervall 6' - - -	15	03 13 23 33 43 53	15	03 13 23 33 43 53
16	- - - - - - - - - - -	16	- - - - - - - - - - -	16	03 13 23 33 43 53	16	03 13 23 33 43 53
17	- - - - - - - - - - -	17	- - - - - - - - - - -	17	03 13 23 33 43 53	17	03 13 23 33 43 53
18	- - - - - - - 35 42 50 57	18	- - - - - - - 35 42 50 57	18	03 13 23 33 43 53	18	03 13 23 33 43 53
19	05 13 23 33 43 53	19	05 13 23 33 43 53	19	03 13 23 33 43 53	19	03 13 23 33 43 53
20	03 13 23 33 43 53	20	03 13 23 33 43 53	20	03 13 23 33 43 53	20	03 13 23 33 43 53
21	03 13 23 33 43 53	21	03 13 23 33 43 53	21	08 23 38 53	21	08 23 38 53
22	03 13 23 38 53	22	03 13 23 38 53	22	08 23 38 53	22	08 23 38 53
23	08 23 38 53	23	08 23 38 53	23	08 23 38 53	23	08 23 38 53
0	20	0	20	0	20	0	20

Am 24.12. Verkehr wie samstags ab ca. 17.00 Uhr Intervall alle 15 Minuten
Am 31.12. Verkehr wie samstags

Auskunft
Wiener Linien: 7909-100
Änderungen vorbehalten

Figure 4-6. Street car schedule of line 37. *Source: Wiener Linien*

The delays presented above are generalized for all types of vehicles. The notion of the two types of delays is shown in Figure 4-6. Two examples are marked on the figure with dashed lines. The circle points to the information about the delay of the same vehicle between two consecutive stops namely Nussdorferstrasse and Glatzgasse. The oval points to the elapsed time between consecutive vehicles arriving at Nussdorferstrasse at a specific day of the week (Friday). This figure depicts a typical schedule posted at each bus and street car stop in Vienna Austria. The schedule of Figure 4-6 differentiates between the days of the week. Where applicable, time scheduling is considered in the planning phase of the trip.

4.6 The point network

The transportation space is conceptualized as a discrete network of points shared by the passenger and the transportation vehicles. Each point is characterized by a set of affordances (Gibson 1986), which are associated with that point. The perception of a sign, e.g. street car line, is an affordance of the sign. The signs which are considered in this research are relevant to moving with public transportation. These signs identify streets and the vehicle's stop, line and destination.

4.6.1 Discrete space

The transportation discrete space is where the spatial wayfinding and business processes are performed. This represents the transportation wayfinding network. Figure 4-7 presents a schematic of this network. The affordances of each point are associated with the operations performed at that point. Points which afford the same set of operations are assigned the same affordance bundle relevant to moving with public transport. For example, all intersections afford to be used for walking, waiting, or looking for wayfinding signs and artifacts. The conceptual model considers the following affordance bundles:

Origin point: the agent affords to walk.

Intersection: the agent affords to walk, to wait, and to perceive

Vehicle stop: the agent affords to walk, to wait, to perceive, to buy a ticket, to validate a ticket, to enter a vehicle and to exit a vehicle

Destination point: the agent affords to walk, to wait, or to stop.

Point of no interest: the agent affords to walk back to his previous point.

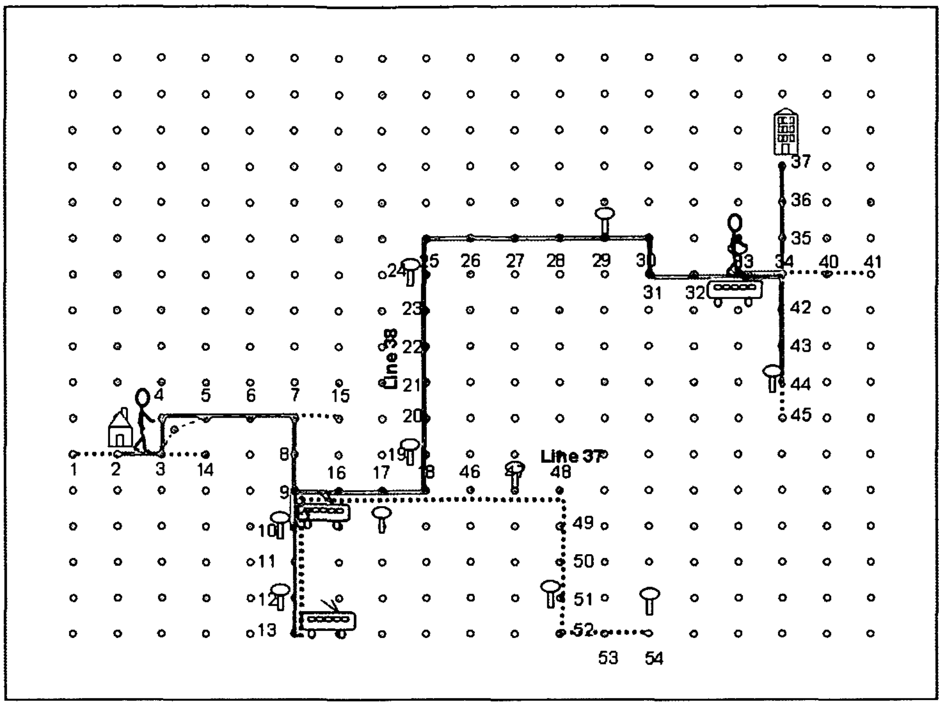


Figure 4-7. Transportation wayfinding network

4.6.2 The passenger in the point network

The passenger perceives the street network and navigates through it using his mental plan. The passenger maintains his state unless an affordance is perceived. When he perceives an affordance, he utilizes an operation to enter a new state. The affordance bundles of the discrete network are unique identification tags. An identification tag bears a variety of representations e.g. pair or street names, the name of a square, the name of a landmark, etc (Winter and Nittel 2003).

4.6.3 The vehicle in the point network

The vehicle follows a predetermined route. Stopping occurs only at vehicle stops. The passenger has no influence on the route of the vehicle. However, the passenger and the vehicle share the ability to change their state at specific locations of the point network, namely at the stops and at the terminals. If an element of the point network is part of the vehicle route and affords to be used for stopping of the vehicle and boarding and exiting passengers from the vehicle, it qualifies as a vehicle stop. The order of stops assists the passenger to determine the stop for getting off the vehicle. The vehicle movement follows the order of the stops and the passenger uses this order as a planning and verification tool in his effort to reach his destination. The latter represents a comparison between the passenger's "knowledge in the head" namely his mental plan (Norman 1988), and the built in "knowledge in the world" (Gibson 1986) in the form of the street car stop sign affordances and hence the street car stop sign order.

4.7 Summary

The conceptual model provides the link between the selected concrete reduced case and the formal implementation. This chapter describes the components which constitute this conceptualization namely, the processes and operations, the passenger, the ticketing, the vehicle, and the point network. It presents the objects and the operations which structure the conceptual model and provides a connection with different levels of detail and different phases of the trip execution. The passenger is the cognitive agent who perceives, plans, decides, acts, and possesses a form of payment resulting in acquiring proof of legitimate use of the transport system. The public transportation vehicle locomotes on a predetermined route based on a fixed schedule. The point network represents a set of discrete points which are shared by the passenger and the vehicle.

The formal implementation of this conceptual model is described in Chapter 7. Chapter 3 provides a theoretic background behind the concepts described in this chapter and Chapter 6 looks at the ontological perspective of these concepts.

5. FORMAL TOOLS AND METHODS

"Since you can't get on and off the train between stops, it doesn't matter what twist the train route takes. What does matter is the essential path and the sequence of stops, with reference to familiar place ... "

Saul Wurman-Information Architects, p15

Wurman's quote refers to the depiction of the Tokyo subway system where the stations are represented as dots on an I-Ching shaped diagram shown in Figure 5-1. Abstraction is used to convey what he thinks is important (Wurman 1996).

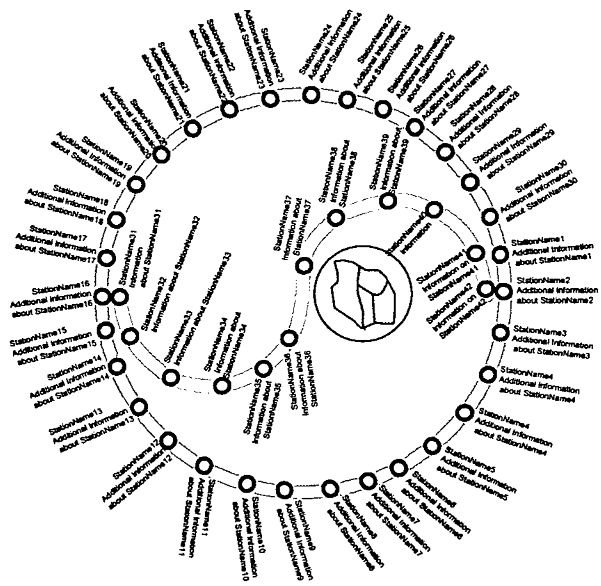


Figure 5-1. Abstract depiction of the Tokyo subway system (Copy from: Wurman, R. S. Information Architects, p.14)

Chapter 4 of this Thesis provided a detailed description of the conceptual scheme which is build based on a reduced case scenario. Figure 5-2 depicts the passage from the observation of the real world to the formal encoding of an abstract case. This chapter presents an overview of the data structures, the formation of algebras for representation of real world phenomena, and finally the functional programming languages. The above tools are used in an effort to formally structure the conceptual scheme. This chapter introduces the state transition diagram and the entity relationship model. Subsequently in Chapter 7, we combine the above knowledge in the elaboration of the formal model.

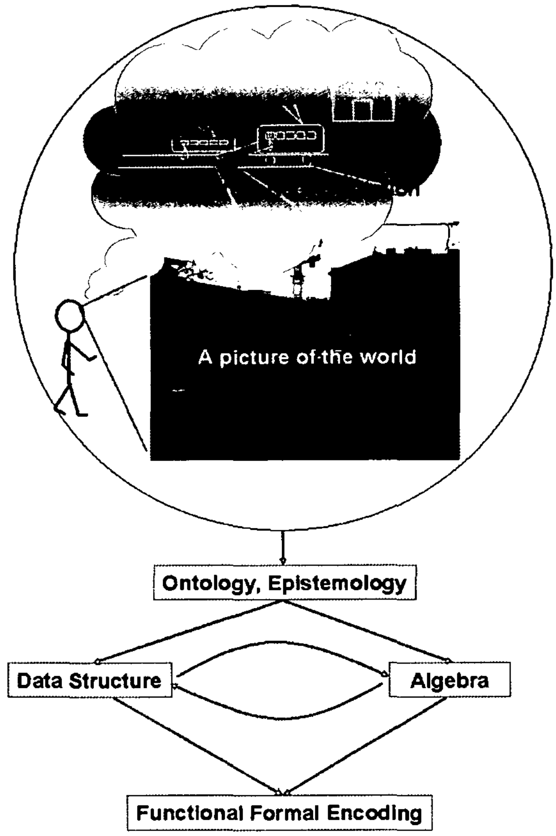


Figure 5-2. From the real world to formal encoding

5.1 Data structure

A data structure is referred to as "*the organization of data and its storage allocation in the computer*" (WordNet 2.0) and as "*bookkeepers that record the state information in a tidy manner so that any part can be accessed and updated separately*" (Nievergelt and Hinrichs 1993). Despite the vast variety of data and applications, there are only few basic data types used to describe them. These data types are: lists, stacks, queues, and dictionaries. These data structures are the building elements of complex structures. Abstraction, classification and aggregation are the three aspects considered when designing data structures. "*An abstract data type consists of a domain from which the data elements are drawn, and a set of operations.*" (Nievergelt and Hinrichs 1993). The abstraction refers to the data types used in the structure and the classification reduces complexity by grouping data together into classes. Generalization refers to the general types which are derived by abstracting common properties of instances (Frank 2005, Submitted) (WordNet 2.0).

A performance assessment is based on asymptotic analysis of time and memory requirements (Nievergelt and Hinrichs 1993). The data structure in this research is presented in Chapter 7 and is not memory or time intensive. The performance aspect is not further considered. The abstraction, classification and aggregation of the data structure are presented in the subsequent chapters.

Nievergelt and Hinrichs propose that the membership property is the most common property attributed to an element in an abstract data structure. This property is often viewed in relation to hierarchy trees. Liskov and Guttag suggest that data abstraction is the most significant step in program design. They list two types of abstraction, namely by parameterization and by specification. Abstraction by parameterization is attained by using parameters while abstraction by specification calls for the operation to be part of the data type (Liskov and Guttag 1986). We observe thus a relation between the data and the algebraic structure as presented further subsequently in this chapter. Liskov and Guttag attribute three important criteria to specifications, namely restrictiveness, generality, and clarity (Liskov and Guttag 1986).

5.2 State transition

Objects have attributes which can change with time. Attributes are expressed with values. The *state* of an object is a set of attribute values for that object at a point in time. A state transition diagram graphically describes the transitions between states. It facilitates the understanding of each state and the identification of missing states. Commonly used elements in the state transition diagram are states, events, transitions, and actions. An *event* is an occurrence which invokes a change in the state of the object. *Transition* is a change of state within the object and *actions* are taken in response to the changes of states. A transition is a passage from state to state.

Matson et al propose that a structural model, a state model and a transition function lay the foundation for all organizational agent models. Their structural model encompasses specific elements such as goals, laws, roles, capabilities, potential achievements, requirements and relationships. The transition function defines the ability to pass from one state to the next state (Matson and DeLoach 2005).

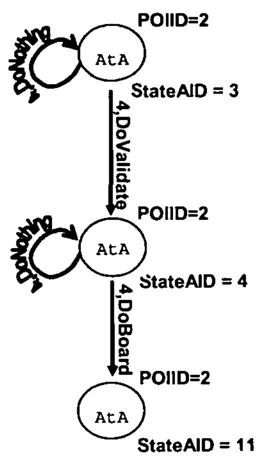


Figure 5-3. Extract of a passenger's state transition diagram

A finite-state automaton is a system which is always in one of a certain number of states. The system moves between states according to instructions (Levy 1993). A cellular model assumes an action space, a set of initial attribute values referred to as initial conditions and a set of rules (Clarke and Gaydos 1998). The cellular model uses the rules to transition between states. Figure 5-3 depicts a portion of a passenger's transition between the states of having a ticket, validating the ticket and boarding the transportation vehicle. Chapter 6 further elaborates on the state transition used to model the reduced case.

5.3 Entity relationship

The entity-relationship identifies the relationships between elements of a data structure.

The two main components of an entity relationship model are the *entities* and the *relationships* amongst them. *Entities* are characterized as the data objects for which information has been gathered. Entities comprise concrete or abstract concepts related to data model. An occurrence of an entity is also known as an instance of that entity. Further details about the entity-relationship model are provided in (Chen 1976).

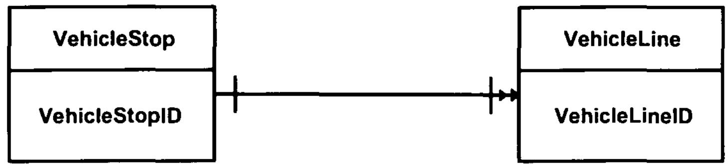


Figure 5-4. Example of a one to one entity relationship

Figure 5-4 presents a depiction of a one-to-many relationship between a vehicle stop data object and a vehicle line data object. Figure 5-5 shows the vehicle line entity table to the left and the vehicle stop entity table to the right. The table between them shows the relationship between the two entities. This is accomplished by using the IDs listed in the two entity tables. In addition to the one-to–many relationship other relationships include: one-to-one; one-to-zero or one; one-to-zero or many, and all combinations thereof. This research uses the entity relationship model in structuring the data of the transportation network discussed in Chapter 7.

VehicleLine
#26: "Hohe Warte"
#27: "DoeblHGatterburg
#28: "GuneschG"
#29: "Grinzing"
#30: "Oberdoebling"
#31: "SilberG"
#32: "GatterburgG"
#33: "HardtG"
#34: "GlatzG"
#35: "NussdorferS"
#36: "Schottentor"

VehicleStop:VehicleLine
#26: #2
#26: #4
#27: #2
#27: #4
#28: #2
#28: #4
#29: #1
#29: #3
#30: #1
#30: #3
#31: #1
#31: #3
#32: #1
#32: #3
#33: #1
#33: #3
#34: #1
#34: #2
#34: #3
#34: #4
#35: #1
#35: #2
#35: #3
#35: #4
#36: #1
#36: #2
#36: #3
#36: #4

VehicleStop
#1: "38S"
#2: "37S"
#3: "38N"
#4: "37N

Figure 5-5. Example of vehicle line and vehicle stop entity tables and their relationship

5.4 The ingredients of an abstract algebra

Algebra is proposed as a tool for abstract modeling (Frank and Kuhn 1995; Winter and Nittel 2003). Algebraic specifications are independent of language (Bittner 2001). Functional programming is used to formalize the algebraic specifications as described in the next section of this chapter. Algebras group objects by common operations that can be carried out on them. This research concentrates on multi-sorted algebras that consist of a set of sorts, operations and axioms. "*The main elements of modeling techniques are categories of types, instances and relationships. Categories share the same attributes relationships and behavior, an instance of a category is a thing that conforms to the definition of the category, a relationship is an association between things or categories*" (Gerbé 1997). Defining categories corresponds to defining partitioning criteria that help to distinguish one instance from another (Gerbé 1997). In the above quote from Gerbé, the term category is the same as class.

Abstraction is a key step in forming an algebra (Frank 1999). Building an effective abstraction scheme of a concrete case leads to an algebra which can be reused. Algebras consist of sorts, operations and axioms (Frank and Kuhn 1995) (Hochmair 2002). The "sorts" are objects of a certain type and they are the input and output of the operations while the axioms are the rules that describe the semantics of the operations (Winter and Nittel 2003) (Hofer 2004). Typically, axioms are independent of implementation. The above definition of sorts and operations conforms with Searle's discussion on objects and acts (Searle 1995).

5.5 Relation Algebra and Relation Data Model

The relation algebra which was introduced by Schroeder (Schröder 1890). Relation algebra is based on relations. A relation's typical representation is:

```
relationName ::
     ID -> value
     ID -> ID
```

A relation has a converse different to the function which does not always have an inverse. Relations can be composed. The components of the relation data model are relations. There are relations which assign an ID to a value and those which assign an ID to another ID. The "to" relation is a relation which assigns an ID to a value or an ID to an ID:

```
"to" relation:
        ID -> value
        ID -> ID
```

The "from" relation assigns a value to an ID or to a set of IDs.

```
"from" relation:
        value -> ID
```

The relation data scheme utilizes two auxiliary functions to convert sets to single value and back: the "Singleton" and "Unique" functions. The input and outputs of these functions are sets of values (Frank 2005, Submitted). The *relation* data model is based on relations while the *relational* data model is based on tables. The "key" of a relational data base table must be unique. Codd elaborates on the distinction between the two models (Codd 1979). The formal relation data model built for this research is described in detailed in Chapter 7.

5.6 Functional programming

The formalization is performed with the use of the functional programming language Haskell. Haskell provides a view of the central ideas for which the model is used. These ideas include abstraction of functions and data types, generalization, polymorphism and overloading (Thompson 1996). There is an extensive literature on the features of Haskell which makes the language suitable for modeling cognitive agents (Bittner 2001; Raubal 2001; Krek 2002; Frank 2003). This section provides a short summary of these features.

5.6.1 Structure of a Haskell Program

A Haskell program has a modular form. From the strict language point of view, *modules* provide a control to namespaces and allow re-use of code in large programs. From the conceptual modeling point of view, modules reflect on the skeletal structure of the model. Modules include *declarations* which define ordinary values, datatypes, type classes, and fixity information. *Expressions* which are the heart of Haskell programming denote values and are characterized by static types. Finally, the *lexical structure* appropriates the representation of Haskell programs in text files (Haskell 1999).

5.6.2 Features of Haskell

In Haskell everything is a function. Functions can be passed to other functions, be the result of a function, stored in a data structure, etc. (Sivashanmugam) . In a functional language new values are built from old ones when expressions are evaluated. Haskell has higher order functions whose arguments or results are functions. The advantage in functional programming is in the evaluation of expressions rather than the execution of commands.

With *strong typing* in Haskell, checking on expressions or definitions is done without performing any evaluation (Thompson 1996). Complex problems can be decomposed in smaller modules. Functions and inference rules describe the relationships between variables. Haskell facilitates rapid prototyping, thus testing at an early development phase of a code.

Winter and Nittel provide the following list of main characteristics summarized above for Haskell: Strong typing, code transparency, polymorphism, modularity, declarative languages, and executable code (Winter and Nittel 2003).

5.6.3 Classes and instances

Algebraic specifications are written in Haskell with classes and instances. Algebras are represented by classes. Classes also describe the behaviour of the data types. The functions describe the operations and the axioms. A class is an assemblage of types over which a function is defined. A class is a structured way to control polymorphism and overloading. Classes allow the declaration of instances within the class and provide definition for overloaded situations related to a class. An instance of a class is a member of that class. A method is a function which can be applicable to any instance which belongs to a class. Classes carry inheritances to subclasses. These classes qualify as super classes (Thompson 1996; Bittner 2001). One such example is the class equality *Eq* which is a super class of the class order *Ord*. *Ord* inherits all operations in *Eq* in addition to its own set of comparison operations.

Haskell allows multiple inheritance since there may be more than one super class to a class. Instance declaration is useful because it assists in describing generic types for functions over a variety of data types, such as lists and trees. This allows functions to work uniformly. The methods and thus the information define a class. Similarly to the module functionality, classes also provide the means for skeletal structure of the model in the formal realm.

Polymorphism in Haskell is a form of overloading which allows functions to be reused. Polymorphism functions in Haskell can be reused with different types of input arguments. The function `length` provides such an example of polymorphism.

```
length :: [Int] -> Int
length :: [Bool] ->Int
etc.
```

5.7 Summary

This chapter presents the formal tools used in this research. Basic data structures are used to complex structures. Abstraction, classification and aggregation are considered when designing data structures. A state transition diagram describes the transitions between states. It assists in understanding each state and identifying missing states. The entity-relationship identifies the relationships between elements of a data structure and primarily consists of entities and the relationships between them.

Algebra is proposed as a tool for abstract modeling (Frank and Kuhn 1995; Winter and Nittel 2003). Algebraic specifications are independent of language. Algebras consist of a set of sorts, operations and axioms. Relation algebra is based on relations. Algebraic specifications are written in Haskell with the use of classes and instances. Classes describe the data types and functions describe the operations and the axioms. This research uses the tools presented here in the formalization of the conceptual model. The algebras which describe an agent's wayfinding with public transportation are described in subsequent chapters.

6. ONTOLOGICAL HIERARCHIES IN A MULTI-MODAL TRIP WITH THE USE OF PUBLIC TRANSPORTATION

> *"...But this is not a stop, said Mr Nixon, in an ordinary sense of the word. Here the tram stops only by request. And since nobody else got off, and since nobody else got on, the request must have come from Watt. ...There is no reason, my dear, said Mr Nixon, no earthy reason, why he should not have requested the tram to stop, as he undoubtedly did. But the fact of having requested the tram to stop proves that he did not mistake the stop, as you suggest. For if he had mistaken the stop, and thought himself already at the railway station, he would not have to request the tram to stop. For the tram always stops at the station. ...*
>
> *...Watt halted before the ticket-window, put down his bags and knocked on the wooden shutter. Go and see what he wants, said Gorman. When Watt saw a face on the other side of the window, he said: Give me a ticket, if you please. He wants a ticket, cried Mr Nolan. A ticket to where? said Mr Gorman. Where to? Said Mr Nolan. To the end of the line, said Watt. He wants a ticket to the end of the line, cried Mr Nolan. Is it a white man? said Lady McCann. Which end? said Mr Gorman. What end? said Mr Nolan. Watt did not reply. The round end of the square end? said Mr Nolan. Watt reflected a little longer. Then he said: The nearer end. ..."*
>
> *(Beckett 1959) Beckett-Watt, p19 and p21*

In the above quote, Beckett describes a scene initially in a tram and later at a ticket counter. His narration, includes operations which are often encountered when moving with public transportation. Beckett decided on the level of detail presented in the scene as suitable for the scope of his novel by including operations such as: identifying a stop when in a moving vehicle or acquiring a ticket to destination. This chapter describes the hierarchical level of detail when looking into such operations. It discusses the ontology of a traveler who is involved in a multi-modal trip with public transportation. Timpf examined and rejected the hypothesis that a passenger's ontology is a subset of the public transport system ontology. She concluded that the two ontologies are not nested but overlap (Timpf 2002). This chapter builds upon the notion that all multi-modal transport actions of a traveler are iterations of basic units of operations namely the wayfinding, business aspect (proof of payment of transport), and container. These operations were introduced in earlier chapters and are analyzed further here. Each ontological unit corresponds to an algebra which includes the plan-know-perceive-decide-act scheme of a cognitive agent discussed in Section 3.5. Some formal notations are introduced which are subsequently used in the formal model.

This chapter utilizes and expands on Frank's 5-tier ontology scheme which was presented in Chapter 3. It is shown how the ontological analysis assists in the formalization of a door-to-door trip with the use of public transportation. The 5-tier ontology scheme (Frank 2001; Frank 2003 Unpublished; Frank 2005-Draft) is utilized to understand the relations between the elements of the units of operations. This research proposes that the ontological decomposition of a traveler's multi-modal trip into processes and operations assists in understanding his ability to perceive and interact with the transportation system.

6.1 Ontological units of public transportation – "Agent", "Move", "Hold", "Contain"

The conceptual model of this research is built upon the notion that a complete trip consists of iterations of basic units of operations of transport (Pontikakis 2004). The granularity of operations was guided by the aim to identify a passenger's minimum set of information needs. The units of operations are connected to the notion of ontological units as described by Frank (Frank 2005-Draft). The operations performed by the passenger during his trip are grouped into these greater units of operations. An "Agent" ontology encompasses the operations of "perceiving" and "deciding". The plan and know components of the agent's structure are part of his mental plan. Figure 6-1 depicts the hierarchies found in the overall task. It distinguishes between the *overall task* of "wayfinding with public transportation", the *processes* of "spatial wayfinding" and "passenger legitimation", the *units* of "move", "contain", and "purchase" and the *sub-units* such as "enter", "exit", and "keep" found in the lower level. Sub-units are aggregated into units and units into processes from the bottom to the top of the ontological structure.

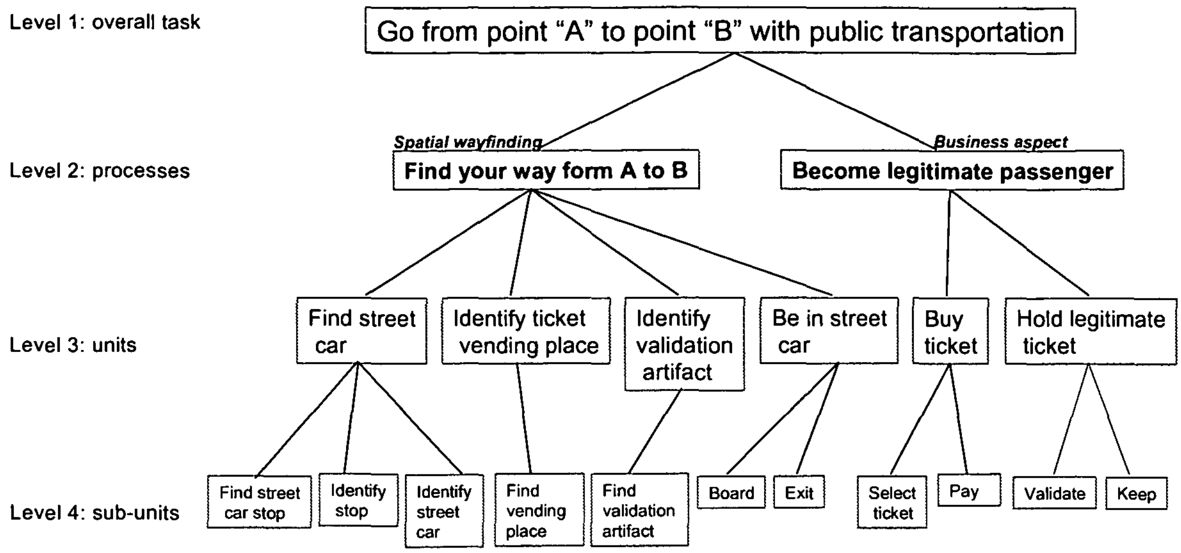


Figure 6-1. Ontological hierarchies

The agent's overall task is to move from A to B legitimately with the use of public transportation. *The hypothesis of this thesis is that the agent performs a wayfinding task with a business aspect embedded in it.* The wayfinding and business ontology is at the top of the hierarchical break down. At a lower hierarchical level, the agent also selects vehicles to enter, and stops to exit (Timpf and Frank 1997). Each ontological unit constitutes a closed system of attributes of operations (Frank 2005-Draft). The operations involved in taking more than one mode of transport and completely executing a trip are repetitions of the basic units of operations.

6.2 Spatial wayfinding process

The spatial wayfinding process takes place when the passenger finds his way from his doorstep to the ticket vending location, from the ticket vending location to the vehicle stop, from one vehicle to another, and from the vehicle stop to the final destination. The passenger utilizes the portion of his mental plan which is in the form of instructions to assure himself of his current position and to reach his next one. The wayfinding issue has been previously addressed by Raubal (Raubal 2001). This research assumes that given a minimum number of interacting components which represent the passenger's knowledge in the head and the knowledge in the world in the form of affordances, the passenger is able to reach his wayfinding goals (Raubal 2001) (Hochmair 2000). The individual wayfinding goals are wrapped in his overall goal of his trip namely to reach his final destination with the use of public transportation. The group of operations of this process forms a "Move" ontology (Frank 2005-Draft), for example the operation `Walk`. The granularity of the operation could be increased to reflect finer moves such as `WalkRight` etc.

6.3 Business process

The business process takes place when a passenger uses the services of a transportation service provider in exchange for a valid payment. The proof of the legitimate use of the transport system is usually a validated ticket. This process constitutes the social and institutional aspect of the overall task, and it is utilized every time the traveler acquires and uses the proof of payment during his trip. The operations involved in this process take place when the passenger purchases the ticket and uses the proof of payment for transport when riding a vehicle.

6.3.1 Buy ticket operation

The agent uses a vending machine to purchase a ticket. The ticket purchasing constitutes a "Hold" ontology with the operation `BuyTicket`. A finer granularity for the `BuyTicket` operation yields `SelectTicket`, and `PayTicketValue` conforming with the numbers 1 and 2 found on the ticket vending machines of the Athens, Greece metro shown in Figure 6-2. The `PayTicketValue` includes a container operation component and is discussed below.

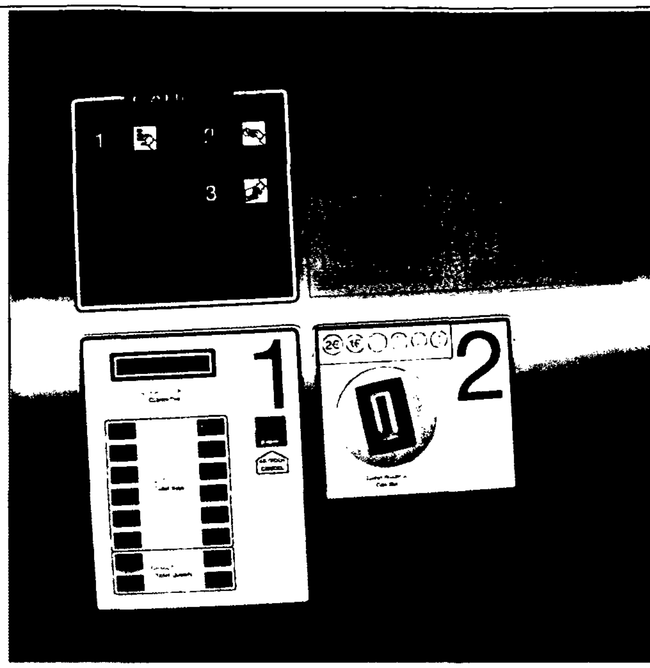


Figure 6-2. Ticket vending machine at Attikon metro in Athens, Greece

6.3.2 Hold legitimate ticket operation

The agent validates his ticket with a validation artifact to become a legitimate user. Ticket validation is a "Hold" ontology with the operation `HoldTicket`. A finer granularity for the `HoldTicket` operation yields `ValidateTicket` and `KeepTicket`.

6.4 Container operations

Container operations encompass the interaction between the passenger and the transportation vehicle and the business transaction concerning the ticket. The following two sub units are part of the container operations.

6.4.1 Pay

These operations which take place when the passenger compares his money supply with the ticket price with the operation `TestPocket` and uses his supply to pay for the ticket with the `PayFromPocket` operation. Added granularity allows the agent to add money to his pocket with the `RefillPocket` operation by withdrawing money from his bank account with the `WithDrawFromBnk` operation.

6.4.2 Select and enter vehicle

These operations take place when the passenger compares the line and destination of a perceived vehicle to his intended line and destination stored in his mental plan. The result is either to wait with the `DoNothing` operation, or enter the vehicle-container with the `Enter` operation. Added granularity, allows him to open the door of the vehicle or ascend the steps.

6.4.3 Exiting vehicle

These operations take place when the agent is in the vehicle, counts the stops after entering, and exits the vehicle-container with the Exit operation. Alternatively, the passenger can enter the vehicle's complement container namely the rest of the world. Added granularity, allows the passenger to request a stop before exiting.

6.5 Structure of operations

Figure 6-3 presents the structure of the processes and operations involved when an agent goes from one place to another with public transportation. Timpf suggests a similar structure for static maps (Timpf and Frank 1997). This research expands on the business component. In the most abstract-coarse level HL1, the agent moves from A to B performing a wayfinding task. In the next level HL2, the agent goes from A to B with public transport, thus utilizing a service. In level HL3, the agent performs finer units of operations when he finds his way to street car stops and vending machines and pays for services. In the next level HL4, the agent performs specific moves as part of his wayfinding tasks such as walking, turning left or right etc. Under HL4 he enters and exits vehicles and acquires and keeps a legitimate proof of payment for the service he is using.

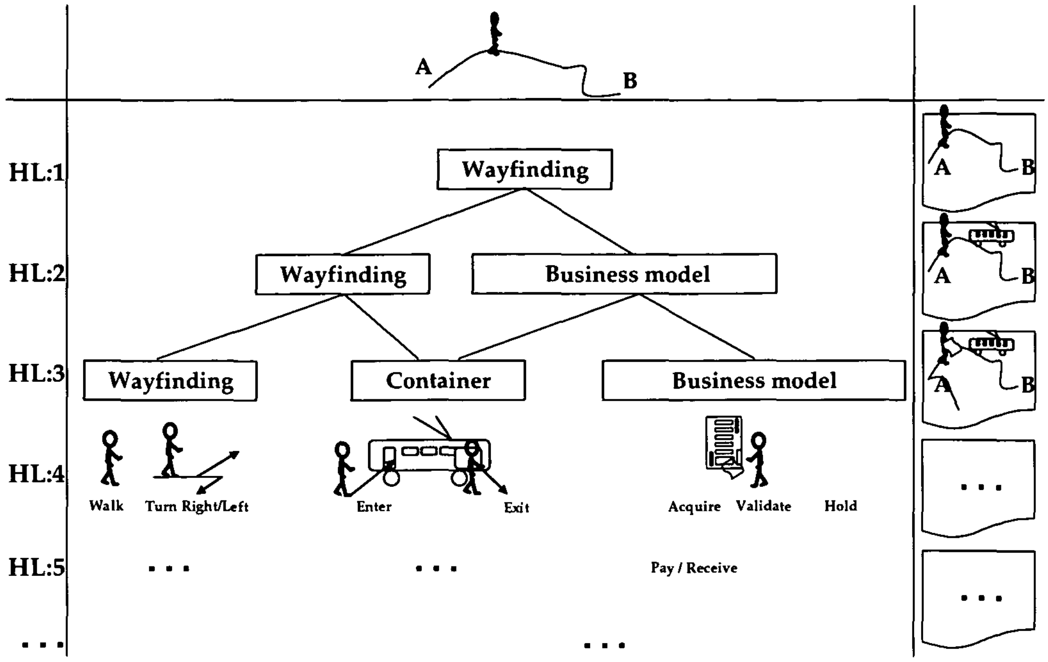


Figure 6-3. Structure of operations

6.6 State transition

The passenger moves through the transportation network by transitioning from one state to another. Chapter 5 introduces the state transition diagram. Figure 6-4 provides the state transition diagram of a passenger who purchases and validates a ticket and uses it for riding one vehicle. The state transition diagram in Figure 6-4 presents the reduced case from Chapter 2.

The numbered `StateAIDs` represent the states of the passenger or the vehicle and the numbered `POIIDs` represent the points in the transportation network. Each arrow in the state transition diagram represents a pair where its first element is the potential next state and its second element is the operation that is needed to achieve this state. The "`DoNothing`" operation is depicted with self closed arrows. The "Move", "Hold" and "Contain" ontological units are embedded in this diagram. For example, in the transition between states 4 and 11 there is a "Contain" ontology, between states 6 and 9 a "Move" ontology, and between 2 and 4 a "Hold" ontology. These examples are marked on Figure 6-4 by dotted rectangles.

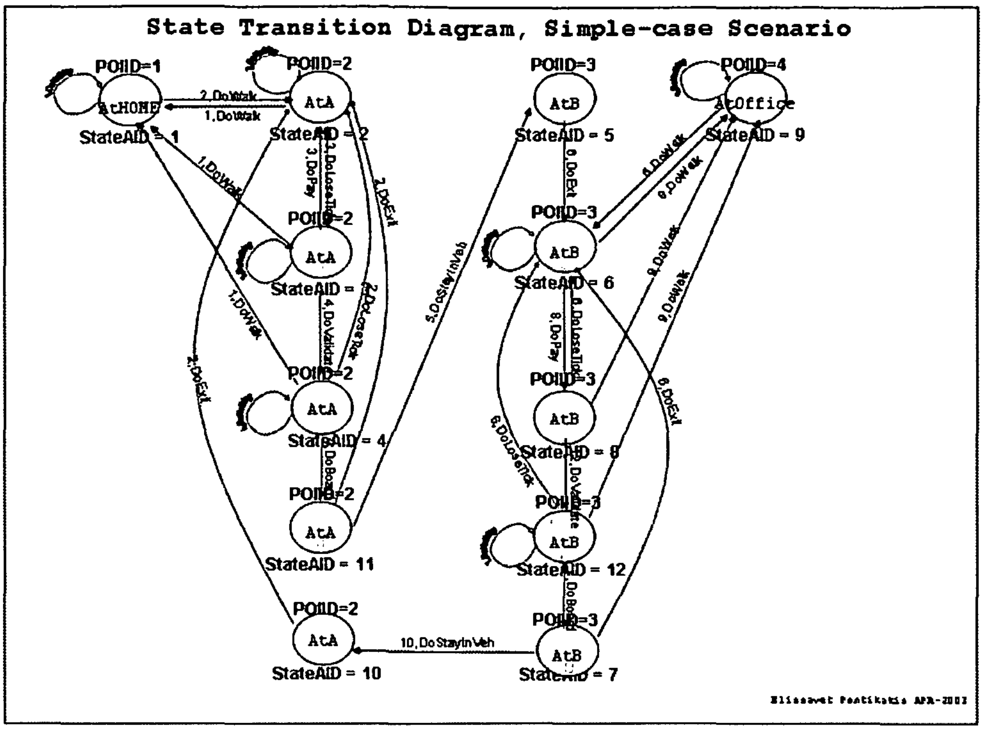


Figure 6-4. One mode of transport in state transitions

6.7 Minimum information needs

Table 6-1 is a table representation of the states in the reduced case. Each state in the state transition diagram corresponds to a row in the table. The first three columns show the start point, the end point, and the operation performed during the state change. The following five columns provide answers to the following questions which define the state of the agent at the end point:

1. Does the agent have enough money to buy a ticket?
2. Does the agent have a ticket?
3. Does the agent have a validated ticket?
4. Is the agent onboard a vehicle?
5. Has the agent reached his destination?

The final column in Table 6-1 shows the ontology which is present during the state change.

Start Point	End Point	Operation	End State					Ontology
			Sum of Money>Ticket Price	Ticket	Validated Ticket	Inside Vehicle	At Destination	
At home	At vehicle stop	Walk	True	False	False	False	False	Move
At vehicle stop 1	At vehicle stop 1	Pay	True	True	False	False	False	Container
At vehicle stop 1	At vehicle stop 1	Validate	True	True	True	False	False	Hold
At vehicle stop 1	At vehicle stop 1	Board	True	True	True	True	False	Container
At vehicle stop 2	At vehicle stop 3	Stay in Vehicle	True	True	True	True	False	Container
At vehicle stop 3	At vehicle stop 4	Stay in Vehicle	True	True	True	True	False	Container
At vehicle stop 4	At vehicle stop 5	Stay in Vehicle	True	True	True	True	False	Container
At vehicle stop 5	At vehicle stop 5	Exit	True	True	True	False	False	Container
At vehicle stop 5	At Destination	Walk	True	True	False	False	True	Move

Table 6-1. Transition states in the reduced case

The information needed to answer the questions that define the state of the agent is shown in Table 6-2. This table provides the minimum information needed to perform the overall wayfinding task.

Question	Information needed
1. Does the agent have enough money to buy a ticket?	ticket price, form of payment
2. Does the agent have a ticket?	need for ticket, ticket vendor location
3. Does the agent have a validated ticket?	need for ticket validation, ticket validation artifact location
4. Is the agent onboard a vehicle?	street car stop location, street car line, and street car destination,
5. Has the agent reached his destination?	street car exit stop, destination location

Table 6-2. Minimum information needs

The agent in the reduced case has no time limitations and no need for time scheduling information. In the cases where a time limitation exists, the time scheduling becomes part of the answer to question 4.

6.8 Interpretation of the 5-tier ontology in multi-modal public transport

The conceptual objects are listed in Chapter 4. The transportation world consists of the passenger, the vehicle, the ticket, and the transportation network, namely everything that is not vehicle, passenger or ticket. This work is presented from the passenger's perspective. This section provides the connection between the tiers of ontology and the elements of the conceptual model.

6.8.1 Tier 0: The physical reality

Little can be said about the physical reality of multi-modal transport. As mentioned earlier, this reality exists; it is dynamic and it is referred to it as the "physical world". This tier encompasses the affordances as found in space.

6.8.2 Tier 1: Observation - Activity

The activity is a major ontological decomposition element. It is related to the ontological unit as described in Section 6.1. People can observe the outcome of these activities. This thesis discusses in Chapter 3 the perception in class of objects. This tier presents no relevance for this research.

6.8.3 Tier 2: Object

We consider the agent, the vehicle, the ticket, and the street network as components of the environment or "world". Time-delay of the dispatched vehicles is modeled as part of the vehicle's schedule. The "world" can become more complicated with the consideration of additional constituents such as ticket validation gates, etc. The agent has a mental plan which contains information acquired through experience and planning. The agent has beliefs concerning his states that include his understanding of his location, and his financial condition, and his possession of a validated ticket. The agent has a final destination stored in his mental plan, and he can identify through perception when he reaches this destination. The vehicle is either in motion or stationary relative to its destination. The "ticket" has a price.

This level of ontology includes all major operations that take place within each operational unit of multi-modal public transport. Formally, they are represented by functions. This serves the generalization idea of this tier. For example, the function "`exit`" and "`board`" should work in different types of vehicles.

In another example, different types of payment are generalized into the class "`Payment`" with a function of an instance of this class "`paying in cash`".

```
class Payments pocket where
payingInCash :: Operation -> pocket -> TicketPrice -> pocket
```

The instantiation of each function is also part of tier 2. For example, the instantiation of the "`payingInCash`" function returns a new sum in the agent's pocket after checking when the operation pay is invoked. Figure 6-5 is an extract of Figure 6-1 presenting an example of the hierarchies found in the business process. This figure also includes examples of instances for the operations "select ticket", "pay", and "validate". The option of paying in cash or credit has been implemented in this research.

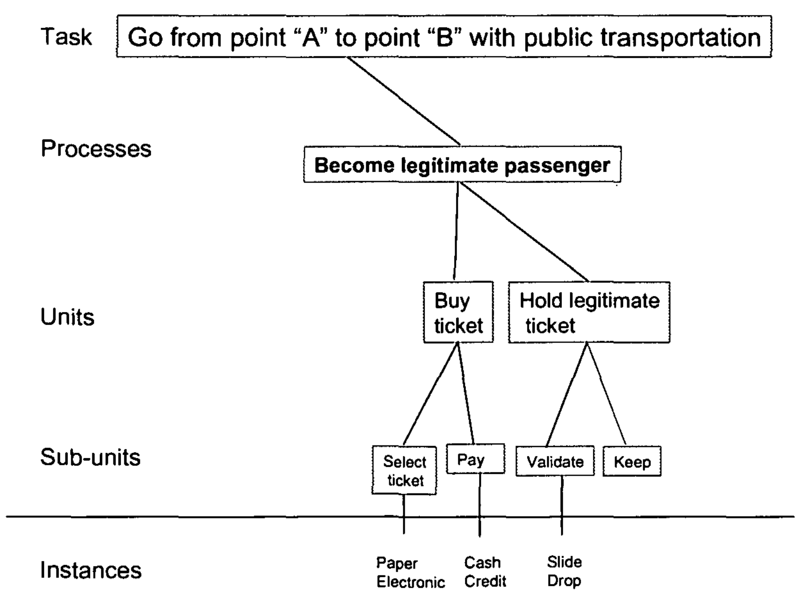


Figure 6-5. Example of hierarchies

Generalization is expressed with the classes. In the structure depicted in Figure 6-5, the sub-units of operations represent classes of operation which generalize the instances. This tier also encompasses cases of aggregation. Aggregation occurs when moving from sub-units to units of operations and from units of operations to processes. For example, the agent is legitimate if he buys and keeps a valid ticket. The process of "becoming a legitimate passenger" is an aggregation of the "buy" and "hold legitimate ticket" units.

6.8.4 Tier 3: Subject- Agent knowledge and goals

This tier covers the agent's knowledge and the goals ("objectives"). Each unit of operations is goal driven with an overall goal superimposed over all other sub-goals. The agent's spatial wayfinding goal is to reach his final destination. The goal of the business process is for the agent to travel legitimately. The overall task is a combination of both goals. The units and sub-units of operations situated below the processes have their individual goals. For example the goal of selecting and entering the vehicle sub-unit is for the passenger to find himself inside the right vehicle. The goal of having a legitimate ticket is to validate and keep the ticket, while the goal of the exiting the vehicle is to identify and exit the vehicle at the right stop.

This tier also encompasses plans and histories (Frank 2005-Draft). The definition of a plan is used here as a list of steps needed to obtain a goal, while the history is a sequence of records used to keep track of the agent's effort to reach a goal. The mental map of the reduced case presented in Table 4-1 shows the hierarchy of the agent's plan.

6.8.5 Tier 4: Institutional/Cultural

The institutional/cultural level of ontology represents the laws that govern each activity. The payment system in a multi-modal transport which depends on the institutional and cultural reality is an area of future research. Proper names used to convey location, destination, and street car line are subjected to law and constitute part of this tier.

6.9 Summary

The ontological decomposition of a traveler's multi-modal trip into processes and operations assists in understanding his ability to perceive and interact with the transportation system. His overall task of "wayfinding with public transportation" is structured in the processes of "spatial wayfinding" and "passenger legitimation". The processes are further structured in the units of "move", "contain", and "purchase" and sub-units such as "enter", "exit", and "keep" operations found in the lower level. The proposed structure of processes and units and sub-units of operations encompass the concepts of generalization (classes), aggregation and instantiation.

The passenger moves through the transportation network by transitioning from one state to another. The identification of the agent's potential states involves answering a minimum set of questions at each point of the point network. The answers to these questions are connected to the *minimum information needs required by the agent to complete his trip legitimately with public transportation*. These are: need for ticket, ticket price, form of payment, ticket vendor location, need for ticket validation, ticket validation artifact location, vehicle stop location, vehicle line, vehicle destination, vehicle exit stop, and destination location. The next chapter presents the formalization of the conceptual model.

7. FORMAL ENCODING OF THE CONCEPTUAL MODEL

> *"...the hack stopped at a way-station and was discharged. Fuller got out and took a seat on a barrow under the awning, as far as he could get from the light; I went inside, and watched the ticket- office. Fuller bought no ticket; I bought none. Presently the train came along, and he boarded a car; I entered the same car at the other end, and came down the aisle and took the seat behind him. When he named the conductor and named his objective point, I dropped back several seats, while the conductor was changing a bill, and when he came to me I paid to the same place—about a hundred miles westward."*
>
> *Twain-The Man That Corrupted Hadleyburg, and Other Essays and Stories, p298*

In the above quote, Twain offers a lively account of a scene at a ticket counter and inside a train. He describes a case where a ticket is purchased while in the transportation vehicle. This chapter describes the formalization of the conceptual model. The conceptual model is discussed in Chapter 4.

7.1 Formalization of the agent's strategy

The conceptual model distinguishes between a spatial wayfinding and a business process. Chapter 6 demonstrates the decomposition of these processes into units of operations. The agent's scheme of "plan", "perceive", "decide", and "act" (Ferber 1998; Raubal 2001; Krek 2002) is formalized through the agent's mental plan and state transition diagram. The *"plan"* component represents the agent's preparation for the trip embedded in his mental plan as list of states. The *"perceive"* component is modeled as affordances of the point network. Each point in the point network is associated with a state in the state transition diagram. The affordance bundle of each point in the point network contains affordances which express potential operations to be performed by the agent at that point. For example, the "perceive" affordance at a point, which represents a street car stop, returns another operation. In regards to the vehicle-container, this operation can be either "board" the vehicle or "wait" at the stop. In regards to the business process, this operation can be "buy" or "validate" a ticket. The *"decide"* component of the agent's scheme models the agent's strategy namely the comparison of the potential next state and its associated afforded operation with the state stored in his mental plan. Finally, the *"act"* component is the operation performed by the agent and represents the outcome of his strategy. The formalization presented here conforms to the structure of operations presented in Chapter 6. The formal code is an adaptation of the algebra presented below.

7.2 The algebra of spatial wayfinding and business process

Figure 7-1 presents the structure of operations during a wayfinding task with public transportation. This figure is a modification of Figure 4-2 where additional emphasis has been added to the lowest level of operations referred to as sub-units in Chapter 6. Algebraic specifications are used for describing the operations (Frank 1999) found at the lower level in Figure 7-1. The first four operations refer to spatial wayfinding. The rest of the operations describe the agent's interaction with the vehicle container and the agent's legitimation with the purchasing and validation of the default ticket of the system.

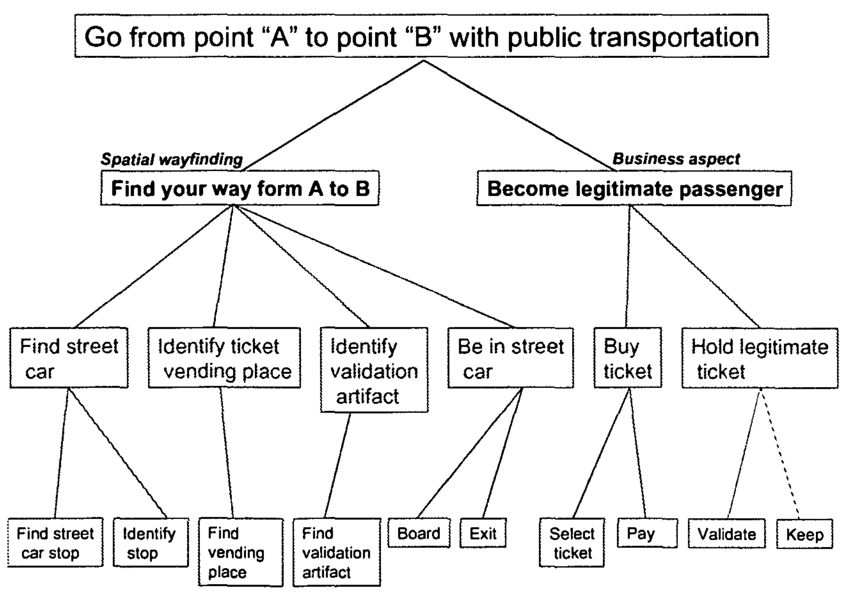


Figure 7-1. Structure of operations

7.2.1 The algebra of spatial wayfinding - the point network and mental plan

This algebra describes the agent's behavior in relation to his mental plan and the point network. POIID refers to the identification of the point in the point network namely, the point of interest ID.

```
class FindNext pOIID where
  findWherePointNet :: pOIID -> PointNet -> pOIID
  findNextPointNet :: pOIID -> PointNet -> pOIID
  findWhereMPlan :: pOIID -> MentalPlan -> MentalPlanEntry
  findNextMPlan :: pOIID -> MentalPlan -> MentalPlanEntry
  findNextOpMPEntry :: pOIID -> Operation -> MentalPlan -> Operation
```

7.2.2 The algebra of the business process

This algebra describes the agent's behavior in relation to legitimately acquiring the services in exchange for a form of payment. It describes the purchasing of the proof for payment in cash or credit and its use through validation. The purchasing constitutes two container algebras (pocket with money and agent holding ticket).

```
class Payments pocket where
  checkPocketForMoney :: pocket -> TicketPrice -> Bool
  payFromPocket :: Operation -> pocket -> TicketPrice -> pocket
  withDrawFromBnk :: BankAccount -> pocket -> TicketPrice ->
      BankAccount
  refillPocket :: BankAccount -> pocket -> TicketPrice -> pocket

class Validates ticket where
  validate :: Operation -> ticket -> ticket

instance Validates Ticket where
  validate op (Ticket pr val) =
      if (op == DoValidate)
           then (Ticket pr True)
      else error "No need for validation"
```

7.2.3 The "container", vehicle algebra

The vehicle: This algebra describes the movement of the vehicle between stops.

```
class VehicleMove pOIID where
  moveOneStop :: pOIID -> Vehicle -> pOIID
```

The algebra of a passenger and vehicle interaction: This algebra describes the passenger's behavior when he is at the bus stop or inside the vehicle.

```
class AgentVehInteracts agent where
  isThisMyBoardingStop :: Operation -> pOIID -> agent -> Bool
  isThisMyBus :: Operation -> Vehicle -> agent -> Bool
  board :: Operation -> Vehicle -> Ticket -> agent -> agent
  isThisMyExitStop :: Operation -> Vehicle -> agent -> Bool
  exit :: Operation -> Vehicle -> agent -> agent
```

7.3 Formalization of state transition

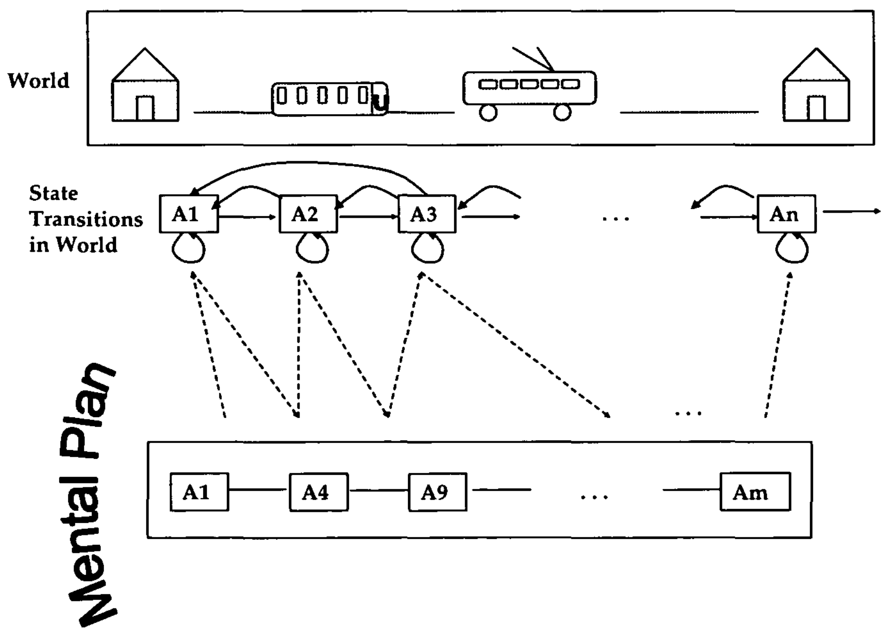


Figure 7-2. State transitions in the world and mental plan

Figure 7-2 is an abstract depiction of an agent's mental plan and the state transitions in the world. The states in the world correspond to potential states of the agent. The solid arrows between the states in the state transition diagram represent a pair where its first element is the potential next state and the second element is the operation that must be performed to achieve this state. The operations required for transitioning from one state to another are related to affordances. These operations are expressions of the knowledge in the world in the state transition diagram.

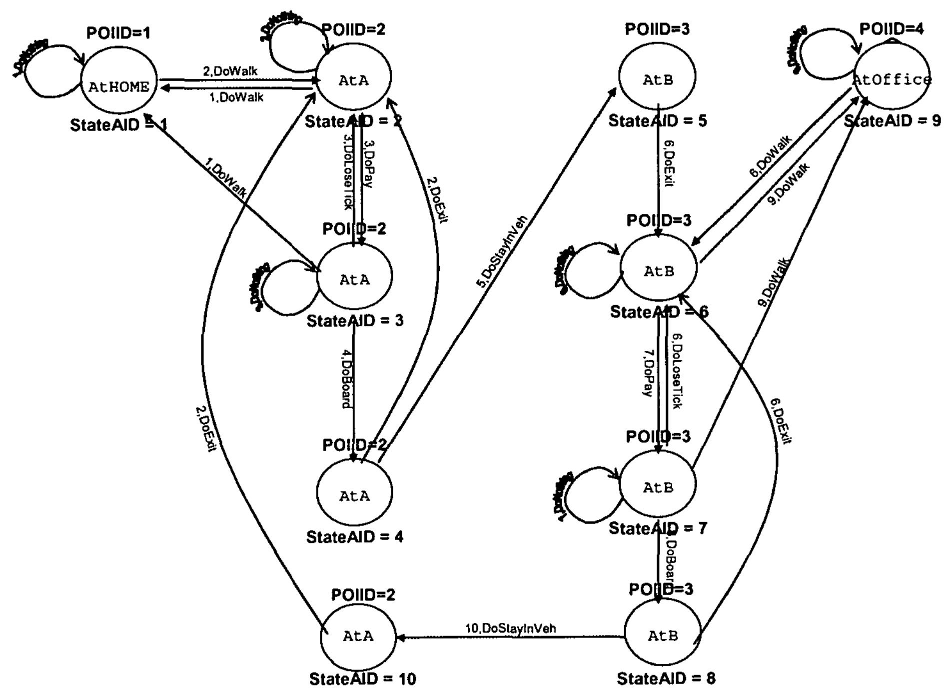


Figure 7-3. State transition diagram of a simplified trip

Figure 7-3 presents the state transition diagram of a trip with public transportation. This state transition diagram is designed to allow for a trip to be made from one end point to another in either direction. In order to simplify the diagram, the "DoPay" operation includes, in one step, both the payment and validation of the ticket. The state transition is formally expressed by the type `stateATransChart` which is a list of all the potential state transitions from each state `stateATrans`. Each `stateATrans` is a pair where the first element is the state ID, `stateAID`, of the current state. The second element of the pair, `PotentStAIDOper`, is a list of all the potential state transitions with the operations required to achieve them from the current state.

```
type StateATransChart = [StateATrans]
type StateATrans = (StateAID, PotentStAIDOper)
type PotentStAIDOper = [StAIDOp]
type StAIDOp = (StateAID, Operation)
```

The outgoing arrows from each state in Figure 7-3 depict the potential new states and the operations required to achieve these states `PotentStAIDOper`. The `StateATrans` for the state with StateAID 2 is shown below as an example. The formalization of the complete state transition diagram of Figure 7-3 is found in Appendix I.

```
stateATrans2 = (2, [(1,DoWalk),(2,DoNothing),(3,DoPay)]) ::
StateATrans
```

7.4 Components of the formal state transition model

The Haskell language supports the modular encoding in the formal model. The structure of the formal modules follows closely the structure of the conceptual scheme. The passenger-point network-vehicle conceptual model is mapped into the `StateTransitionWorld`, `Passenger`, `PointNet` and `Vehicle` Haskell modules. The units of operations are encoded within the `StateTransitionWorld` module.

7.4.1 The StateTransitionWorld module

The `StateTransitionWorld` module is comprised of the `Passenger`, `PointNet` and `Vehicle` modules and plays the role of the data type integrator.

```
module StateTransitionWorld where
import Passenger
import PointNet
import Vehicle
```

7.4.2 The Passenger module

The `Passenger` module encodes the cognitive agent, namely the passenger. According to the conceptual description, the agent has a state, possesses a mental plan, a form of money, and a ticket and he is either in or out of a transportation vehicle. The agent's state provides a connection with the agent's position and thus the affordances perceived at that position. Figure 7-4 presents a schematic of this conceptual structure.

The Passenger module covers the agent's mental plan and his spatial and financial state. The agent's cognition of his state in association with him being in or out of a vehicle container is modeled through the Boolean type `InVehicle`.

```
data Agent = Agent StateAID MentalPlan Pocket Ticket
InVehicle
```

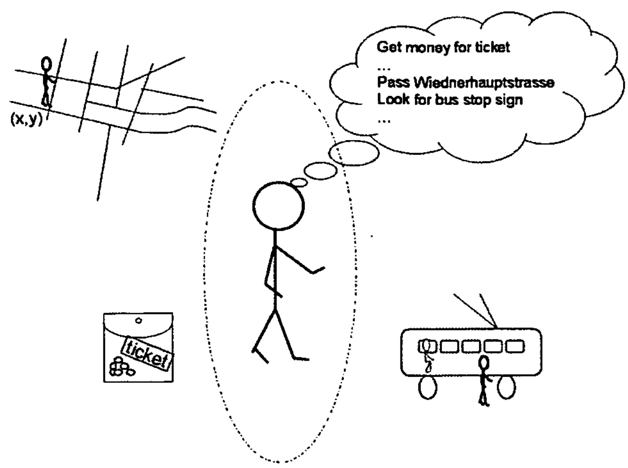


Figure 7-4. Agent's schematic of formal components

The mental plan formalizes the agent's *knowledge in the head*. The knowledge in the head represents the agent's experience and planning efforts. The data type `MentalPlan` models the agent's list of states stored in his head which are needed to complete his trip. These states are expressed with an identification number namely `StateAID` and are associated with operations in the state transitions in the world. The operations are expressions of the needs for information about the agent's spatial and business aspect of his trip as described in Chapter 6. Typically, transportation is a service requiring payment and the passenger has this information from experience. The points of origin and destination are included as states in the `MentalPlan`.

```
type MentalPlan = [StateAID]
```

The agent starts at his origin and before he moves to a new state, he looks into the mental plan and identifies his next planned state. He then looks into the state transitions in the world and selects the pair of the potential state and operation, which takes him to the next state in his mental plan. The spatial wayfinding piece has already been extensively modeled through previous work (Raubal 2001) and is simplified in this formalization. For the scope of this research, the agent's mental plan provides instructions on how to locomote successfully between states until he perceives affordances of public transport related signs. Acquisition of a valid proof of payment for transport follows and completes the business aspect of the trip. Subsequently, his mental plan instructs him to move to a state for boarding a public transport vehicle with "x" destination and for exiting at a vehicle stop "y".

The two sample mental plans shown below transition the agent among states found in Figure 7-3. The operation for transitioning to these states include walking to the vehicle stop (planned knowledge), purchasing a validated ticket at a vehicle stop (knowledge from experience), boarding the vehicle at the vehicle stop (planned knowledge), staying in the vehicle until the appropriate exit vehicle stop (planned knowledge), exiting the vehicle (planned knowledge) and walking to the destination (planned knowledge). The state IDs in `mentalPlan1` and `mentalPlan2` conform to the `stateAIDs` found in Figure 7-3:

```
mentalPlan1 = [1, 2, 3, 4, 5, 6, 9] :: MentalPlan
mentalPlan2 = [9, 6, 7, 8, 10, 2, 1] :: MentalPlan
```

`mentalPlan1` takes the agent from a beginning state to an end state while `mentalPlan2` is the return trip. As shown on the state transition diagram and encoded in Appendix I, if the agent is unable to transition to the states with `stateAID` = 3 or `stateAID` = 7, his wayfinding task fails. This can happen either because he can not successfully execute the payment operation or because the states with `stateAID` = 3 or `stateAID` = 7 are not included in his mental plan. The transition to these two states is a business process. This knowledge is formally embedded into the state transition diagram.

All of the other states refer to the spatial wayfinding. The knowledge in the world for spatial wayfinding is also embedded in the state transition diagram. If this knowledge is inadequate or if the agent's mental plan does provide enough or correct information, the spatial wayfinding task fails. Thus the overall wayfinding with public transportation can not be accomplished.

The agent checks his financial state before he leaves the origin. The `checkForTicket` and `checkPocketForMoney` operation returns an agent who knows whether he has already a ticket or not and whether he is able to make the payment needed for his trip with public transportation. This initiates the payment process of the overall wayfinding task.

The data type `Ticket` codes the agent's legitimation status as a user of the public transport system with the possession or not of a validated proof of payment.

```
data Ticket = Ticket HaveTicket TicketPrice Validated
type HaveTicket  = Bool
type TicketPrice = Int
type Validated   = Bool
```

The data type `Pocket` codes the agent's financial condition and his possession of a ticket.

```
data Pocket = Pocket Money Ticket
```

Money is modeled in the form of cash or credit, as in credit cards, both of type Integer.

```
data Money = Money Cash Credit
type Cash = Int
type Credit = Int
```

7.4.3 The `makeTrip` operation

The agent's mental plan is a list of states. The agent starts at his origin and subsequently uses his mental plan to identify his next state with the function `findNextMPlan`.

```
findNextMPlan :: stateAID -> MentalPlan ->stateAID
```

He then looks into the state transitions in the world and selects the arrow (Figure 7-2 and Figure 7-3), namely the pair of the potential state and operation, which takes him to the next state in his mental plan with the functions `findNxtLnInChart` and `findStAIdOper`.

```
findNxtLnInChart :: StateAID -> MentalPlan -> StateATransChart ->
PotentStAIDOper

findStAIdOper :: StateAID -> PotentStAIDOper -> StAIDOp
```

The agent makes one step of his trip with the `makeOneStep` function which returns the appropriate pair of state and operation. The `make1StepNewAgent` function supplies the above state to the agent, performs the operation and returns a new agent who has either reached his destination or he needs additional transitions through states to do so. The `TicketPrice` is used to identify if the agent has enough money to pay for the ticket.

```
makeOneStep :: Agent -> TicketPrice -> StateATransChart -> StAIDOp
make1StepNewAg :: Agent -> TicketPrice -> StateATransChart -> Agent
```

The `makeTrip` function recursively calls the `make1StepNewAgent` function to perform the complete trip.

```
makeTrip :: Agent -> TicketPrice -> StateATransChart -> [Agent]
```

The formalization of the state transition diagram shown in Figure 7-3 is found in Appendix I. The information needed for making a complete trip is the collection of information needed by the agent to move from any one state to the next. Chapter 6 provides a list of the minimum needs.

If public transportation was a service free of charge for the passenger, the ticketing aspect in the formalization of the state transition diagram could be eliminated. By removing the states which are relevant to ticketing, only the spatial wayfinding aspect would remain. Typically, however, public transportation is a service requiring payment. The simulation of the state transition diagram demonstrates that the overall wayfinding task requires that both the spatial wayfinding and the business components are necessary for wayfinding with public transportation.

The hierarchical breakdown of the business aspects has been presented in detail throughout this thesis with special emphasis given in Chapter 6. Chapter 6 provides a description of the ontology of a trip with public transportation. The agent's location is an ID of the point network data base which has been independently formalized in a Haskell relation database described below and listed in Appendix II.

7.5 The relational model of the knowledge in the world-PointNet module

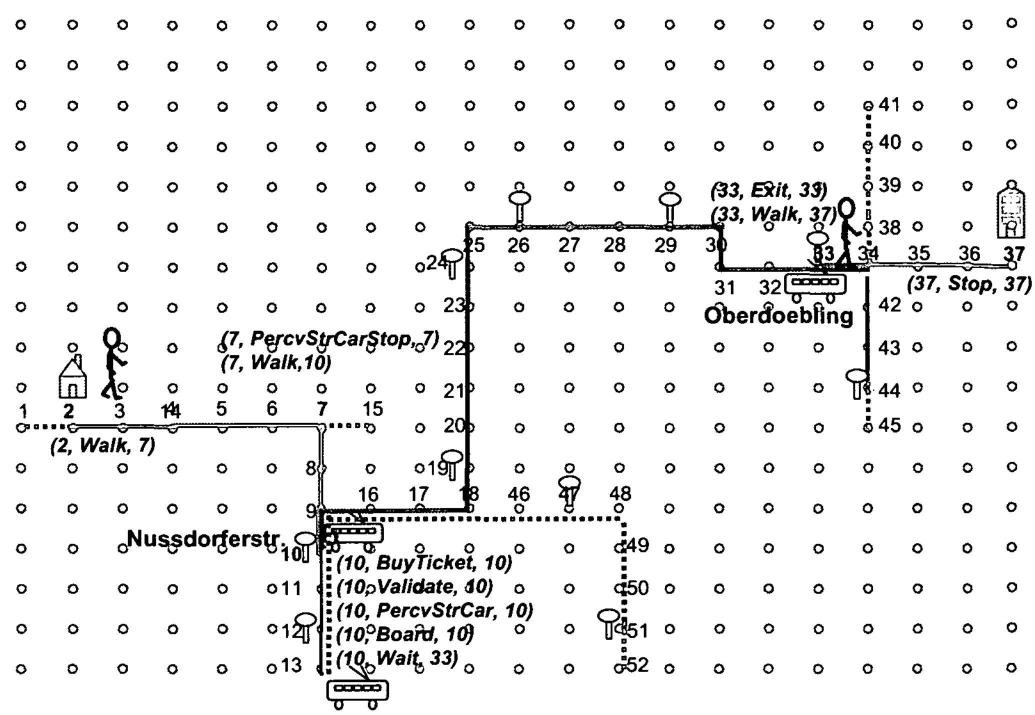


Figure 7-5. Discrete Formal Network

The `PointNet` module encodes the fixed point network that is shared by the passenger and the vehicle. The information embedded in the point network is the knowledge in the world for the reduced case. The data structure of points and operations is a relation database (Frank 2005, Submitted). Chapter 6 described how a relation database is constructed. The point network consists of an assemblage of discrete points of interest. Each point is characterized by an ID and a set of operations that it affords, namely a set of affordances, alternatively called an "affordance bundle".

Figure 7-5 is a schematic view of the point network used for formalizing the knowledge in the world for the reduced case. The IDs of the points in the representation of the physical world range from 1 to 52. They are represented in the relation model by the data type `pointData` of type `Int`. Each value of the `pointData` uniquely specifies the point and the line and direction in the cases of vehicle stops. The values of `pointData` map to the values in Figure 7-5 in the following way in order to facilitate debugging and as an aid for the reader: the first three digits of the each `pointData` is the integer yielded by adding 100 to the value in Figure 7-5. The fourth digit of the value of `pointData` indicates if the point is a vehicle stop. This mapping has no significance for the structure of the relation model and the values can be treated as arbitrary IDs. The formal code of the relation model is found in Appendix II. The model in Appendix II includes the entities needed for the relation model of the knowledge in the world for the reduced case and additional entities for future research.

IDs of point network: The points of interest are internally associated with elements of the transportation network such as vehicle stops, lines, destinations, and routes. The model includes only the points which are of interest to the aim of this research shown by the numbered dots in Figure 7-5. The "point of interest identification" number, `POIID`, is an integer and is used as a reference to the point of interest.

`FixedType(s)` of the point network: The relation data model for the point network is composed of different "`FixedType`" data which are used are listed below:

```
data FixedType =
        BusStopT          |        --Bus stop
        BusStpOrdT        |        --Bus stop order
        BusLnPntT         |        --Bus line to point
        BusDestT          |        --Bus destination
        BusLineT          |        --Bus line
        AffordanceT       |        --Affordance
        AffrdBundleT      |        --Affordance bundle
        AffrdBPntT        |        --Affordance bundle to point
        AffrdToBndlT      |        --Affordance to affordance
                                   --bundle
        BusStopLineT      |        --Bus stop to bus line
        DelayT            |        --Delay
        OrderT            |        --Order
        PointT            |        --Point
        BusIDT            |        --Bus ID
        BusIDLn                    --Bus ID to line
```

The data structure follows closely the above scheme containing the following types:

```
type BusStop = String            --Bus stop name
type BusLine = String            --Bus line name
type BusDest = String            --Bus destination name
type Affordance = String         --Affordance
type AffrdBundle = String        --Affordance bundle name
type AffrdToBndl = String        --Affordance to affordance
                                 --bundle (relationship)
type POIID = Int                 --Point
type Order = Int                 --Order
type Delay = Int                 --Delay from terminal to stop
type BusStpOrd = Int             --Bus stop order
type BusLnPnt = Int              --Bus line to point
type AffrdBPnt = Int             --Affordance bundle to point
type DelayBStpNB = Int           --Delay to bus stop
type BusID = Int                 --Bus ID
type BusIDLn = Int               --BusID to bus line
```

The point network relation database OODB is built to contain the above types either as values or as relevant IDs as described in Section 5.5 and as listed below. The OODB produces relations between the point network and the objects and operations.

```
data OODB = OODB {lastid::ID,
            name :: RelVal Name,
            busStop :: RelID,
            busLine :: RelID,
            busDest :: RelID,
            affordance :: RelID,
            affrdBundle :: RelID,
            affrdToBndl :: RelID,
            point :: RelVal POIID,
            order :: RelVal Order,
            delay :: RelVal Delay,
            busStpOrd :: RelID,
            busLnPnt :: RelID,
            affrdBPnt :: RelID,
            delayBStpNB :: RelID,
            busID :: RelVal BusID,
            busIDLn :: RelID,
```

```
fixedType :: RelVal FixedType}
```

Relation paths: The identification of the semantic value of an entity of interest is achieved by recursively accessing the relationship paths (Nievergelt and Hinrichs 1993). The "`identify`" function is used to return a unique internal identifier from a semantic value while and it is a form of the "`from`" function. The "`to`" function returns a semantic value from an internal identifier. The "`from`" function returns an identifier from another identifier of a related entity (Frank 2005, Submitted). For example in our run, `point 1172` has the internal `ID: #96,` it is of `FixedType PointT`, it is associated with `#6` in the `BusLnPnt:BusLine` relation `(#99: #3)` which in turn expresses its relation with a bus line assigned to destination with internal `ID: #3` to `"Grinzing"` through the relation `BusDest:BusLine (#29: #3)` etc.

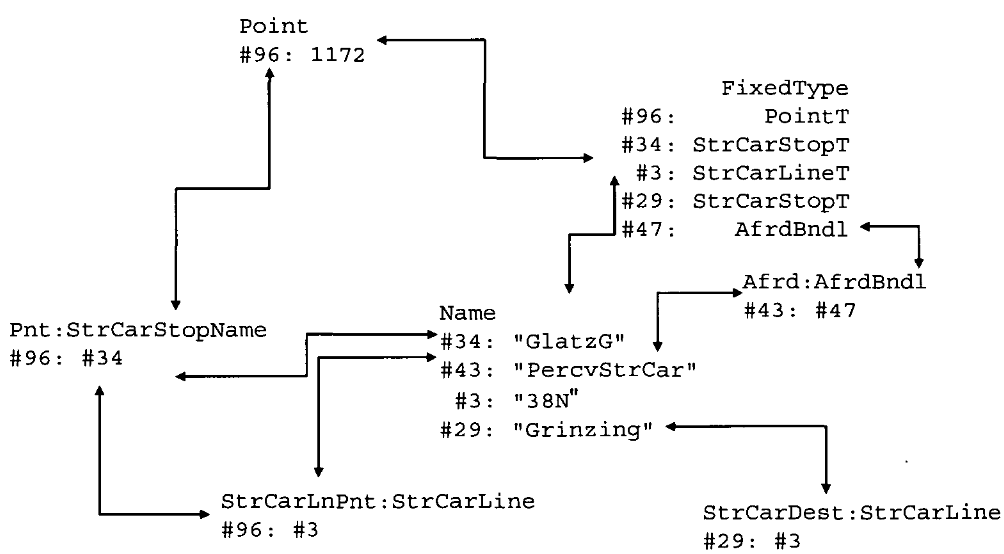


Figure 7-6. Relation trace path (example for point 1172)

Figure 7-6 provides and example of tracing relations. The entity relationship diagram shown in Figure 7-7 is for one street car line going in one direction. A single arrow represents a one-to-one relationship and a double headed arrow a one-to-many relationship. Each street car line has a one-to-one relationship with the street car destination table. Each street car line has a one-to-many relationship with the street car stops. Each street car stop has a one-to-one relationship with the street car point table. Each entry in the street car point table has a corresponding entry in the point table. Each affordance has a one-to-one relationship with the affordance bundle table. Each entry in the point table represents a point in the transportation network used in this model. Each point in the point network has a one-to-one relation with the affordance bundle table.

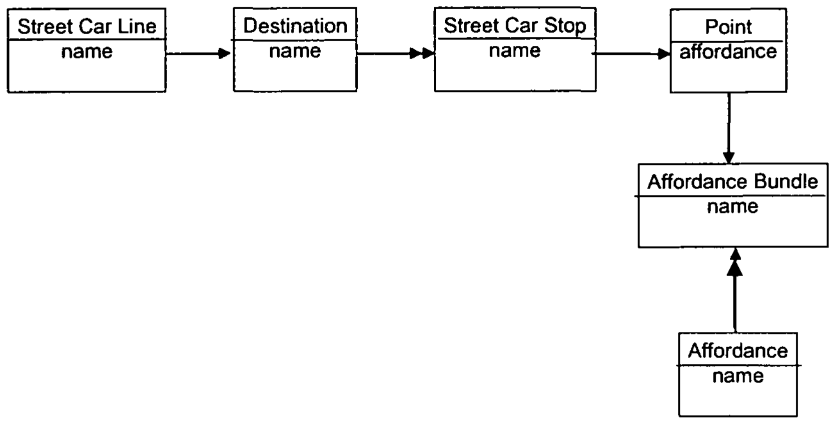


Figure 7-7. Entity relationship diagram of the transportation point network

7.5.1 Initialization values

The initialization values of each data type are derived from the reduced case.

```
busStopData =
[("NoLabel"),("Schottentor"),("NussdorferS"),("GlatzG"),
("HardtG"),("GatterburgG"),("SilberG"),("Oberdoebling"),
("Grinzing"),("GuneschG"), ("DoeblHGatterburgG"),
("Hohe Warte")] :: [(BusStop)]

busLineData =
[("37N"),("38N"),("37S"),("38S")] :: [(String)]

busDestData =
[("Hohe Warte"), ("Grinzing"), ("Schottentor")] ::
[(Name)]
```

The `BusStop`, `BusLine`, `BusDest` are perceivable signs. The signs employed that are of interest to the passenger's needs are the signs indicating the vehicle's stop "`BusStop`", the vehicle's line "BusLine" and the vehicle's destination "BusDest".

Sign label (SignLabel) is the label of the sign related to its context and content. A discussion about signs is included in Chapter 3. The sign labels are implemented as names in the database. For example, "Glatzgasse" is the name of a bus stop on tram line "38" going to "Grinzing".

Vehicle schedule: The bus lines are constructed within the relation data framework including the order of the bus stops and the schedule of the buses. The schedule is an entity of the PointNet data structure mapped by the time delay needed for a vehicle to reach a stop. A standard delay for dispatching vehicles at the terminal is used. Each vehicle dispatched has a unique ID and it is associated with a bus line through the VehIDLn:BusLn relation.

```
orderData =
[0..20] :: [Int]
delayData =
[0..10] :: [(Delay)]
vehIDData =
[100..499] :: [Int]
vehIDLnData =
[(100,"37N"),(101,"37N"),…,(200,"38N"),(201,"38N"),…
(300,"37S"),(301,"37S"),…,(400,"38S"),(401,"38S"),…] ::
[(VehID, BusLine)]
```

The relation model includes the street car time delay as units of time with unequal length. Each vehicle is related to a line and to a time delay from terminal. The schedule at a vehicle stop is a list of pairs. The first element of each pair is the vehicle ID and the second element is the time delay from terminal [(VehID, Delay)]. For example, the street car stop "HardtG" has a 5 minutes delay from the terminal. If the first vehicle is first dispatched at 5:00 am or at 300 min., the first vehicle will depart at 305 min. at "HardtG" :

```
[(200, 305), (201, 315), (202, 325), …,
(vehID,(vehID - head [vehID])*10 + delay)]
```

The formula vehID - head [vehID])*10 + delay yields the time that a vehicle reaches a specific vehicle stop as a function of the vehicle ID and the delay which expresses the time that is required by a vehicle to reach the vehicle stop.

Affordances of the point network: An affordance bundle, `AffrdBundle`, is associated with the points of the point network. The affordance bundle denotes the affordances perceived by the agent at a specific point such as "`AtIntersection`" or "`AtBusStop`" etc. For the data model, an affordance bundle is a name tag for the point. An affordance bundle is modeled as follows:

```
affrdBundleData =
[("AtNoInterest"),("AtHome"),("AtIntersect"),
("AtBusStop"),("AtDest")] :: [(AffrdBundle)]
```

The affordance bundle associates each point in the point network to operations related to perception and human cognition and operations pertinent to moving by public transportation. "`AtNoInterest`" is an affordance bundle assigned to a point presenting no interest regarding moving by public transportation.

Affordance, `Affordance`, is what each point perceptually conveys to an agent about what it can be used for (Gibson 1986). The relation model includes the following affordances. They are related to the operations discussed in Chapter 6.

```
affordanceData =
[("Wait"),("Walk"),("PercvStrCar"),("PercvStrCarStop"),
("BuyTicket"), ("Validate"),("Board"),("Exit")] ::
[(Affordance)]
```

7.5.2 The Vehicle module

The `Vehicle` module encodes the transportation vehicle. Following its conceptual depiction, the vehicle encompasses an ID, a route, a location and a state of motion.

```
data Vehicle = Vehicle VID VehRoute MoveState VLocation
type VID =  VehID
type VehRoute    = [POIID]
type VLocation = ID
```

The `VehRoute` is a list of points which express the street car stops of the route. The list is ordered based on the bus stop order within a bus line.

7.5.3 The Operations module

The `Operations` module formalizes the concept of operations as described in Chapter 6. The wayfinding and business ontology is at the top of the hierarchical ladder. At a lower hierarchical level, the agent also selects vehicles to enter, and stops to exit (Timpf and Frank 1997). The operations involved in taking more than one mode of transport and completely executing a trip are repetitions of the basic units of operations. The operations which are explicitly expressed in this module are associated with the affordances stored in the point network.

Wayfinding: The spatial wayfinding operations move the agent in a predefined path until he perceive transportation related signs such as bus stops or business wayfinding affordances such as ticket vending machines or validation gates. The formalization of the wayfinding unit simulates the agent's wayfinding behavior.

```
class FindNext pOIID where
  findWherePointNet :: pOIID -> PointNet -> pOIID
  findNextPointNet :: pOIID -> PointNet -> pOIID
  findWhereMPlan :: pOIID -> MentalPlan -> MentalPlanEntry
  findNextMPlan :: pOIID -> MentalPlan -> MentalPlanEntry
  findNextOpMPEntry :: pOIID -> Operation -> MentalPlan -> Operation
```

The business operations: The payment and ticket schema formalizes the semantics of the institutional processes for acquiring and handling the proof of payment of transport by the passenger. The integration of the business wayfinding aspect to the spatial wayfinding constitutes the novelty of this research. Container operations are present in this process when adding or withdrawing money from the pocket of from the bank account.

```
class Payments pocket where
  testPocket :: pocket -> TicketPrice -> Bool
  payFromPocket :: Operation -> pocket -> TicketPrice -> pocket
  withDrawFromBnk :: BankAccount -> pocket -> TicketPrice ->
      BankAccount
  refillPocket :: BankAccount -> pocket -> TicketPrice -> pocket


instance Payments Pocket where
  testPocket pockt tckPrc =
           if (pockt >= tckPrc) then True
```

```
              else False

      payFromPocket op pockt tckPrc=
              if (op == DoPay)
                  then if (testPocket pockt tckPrc == True)
                        then pockt - tckPrc
                  else error "Not enough money in pocket"
              else pockt

      withDrawFromBnk bnkAcc pockt tckPrc =
              if (testPocket pockt tckPrc == False) then bnkAcc'
              else bnkAcc
                  where
                        bnkAcc' = bnkAcc - tckPrc

      refillPocket bnkAcc pockt tckPrc =
              if (testPocket pockt tckPrc == False) then pockt'
              else pockt
                  where
                        pockt' = pockt + tckPrc

   class Validates ticket where

        validate :: Operation -> ticket -> ticket


   instance Validates Ticket where

        validate op (Ticket pr val) =

                if (op == DoValidate)

                        then (Ticket pr True)

                else error "No need for validation"
```

Boarding and exiting: The agent perceives the bus stop, and the upcoming vehicle. He decides to board the vehicle or to wait for the next one. A container operation is then performed. His selection and interaction with the vehicle for boarding is based on the perception of the transportation related signs and the mental plan entry.

```
--spatial wayfinding
class AgentContainerInteracts agent where
   isThisMyBoardingStop :: Operation -> pOIID -> agent -> Bool      ...
       if (head (to oN name (to oN busLnPnt (from oN point
             [poi]))) == busLine) then True
       else error "Walk to next in point net"
       isThisMyVeh :: Operation -> Vehicle -> agent ->
             Operation
...
             if  fst( head mVSchedul == busLine ) then
                  if (aTime <= snd (head mVSchedul)) then "Board"
             else "Wait"

--container
  board :: Operation -> Ticket -> agent -> agent
...
  board op (Ticket tckPrc val) (Agent poi (mPl:mPls) pkt)
              = if (val == False) then error "Validate"
       else (Agent poi mPls pkt)

--spatial wayfinding
isThisMyExitStop :: Operation -> Vehicle -> agent ->
  Boolclass ExitVeh agent where
...
             if (vStop == head mPls) then True
             else isThisMyStop op (Vehicle stV vStops) (Agent
                  poi (mPl:mPls) pkt)

--container
exit :: Operation -> Vehicle -> agent -> agent
...
  if ((isThisMyStop op veh (Agent poi (mPl:mPls) pkt)) ==
        True) then
             (Agent poi mPls pkt)
       else error "Wait-Do not exit"
```

7.6 Summary

The cognitive agent is formalized based on the "plan", "perceive", "decide", and "act" scheme. Planning is formally coded by the agent's mental plan. Perception is modeled as affordances of the point network. The decision models the agent's strategy, namely the comparison of the potential next state and its associated afforded operation with the state stored in his mental plan. Finally, the actions are the operations performed by the agent and represent the outcome of the agent's strategy. The formalization presented in this chapter conforms to the structure of operations presented in Chapter 6. The formal code is an adaptation of the algebraic specifications of the operations. There are operations which express the spatial wayfinding aspect of the trip and those which express the business transactions. This finding conforms to the hypothesis of this research which is that a passenger's wayfinding trip with public transport is a composition of a spatial wayfinding and a business component.

The passenger moves in a world which allows transitioning from one state to another. The agent's mental plan models the information needs in the form of knowledge in the head resulting from planning and experience. The state transition diagram models the knowledge embedded in the world. The information needed for making a complete trip is the collection of the information needed by the agent to move from any one state to the next. When there is not enough information to move between any two consecutive states in the state transition diagram, the overall task can not be accomplished. This chapter also includes a comprehensive description of the formalization of the knowledge in the world as a relation model for the reduced case.

8. CONCLUSIONS AND FUTURE RESEARCH

This chapter summarizes the work presented in this thesis. It describes the findings of this research and concludes with a discussion on future work.

8.1 Review of research

The aim of this research was to assess the information needs for the business transactions and physical movements of a passenger who uses public transportation. This research was motivated by the two case scenarios conducted for the Bundesministerium für Wissenschaft und Verkehr in Vienna Austria aiming in identifying the missing components of the information chain in public transport. Chapter 2 presented the two motivating scenarios and the reduced case used in developing the conceptual model of this research.

Chapter 3 elaborated on the theories used in this research when translating the conceptual into formal model. At first, it discussed the cognitive agent theory and the multi-agent systems to structure the agent's "plan", "perceive", "decide", and "act" scheme. Subsequently, it presented the theory of perception and the knowledge found in the world in the form of affordances and image schemata. This chapter also discussed the theory of signs which explains the agent's ability to perceive a specific sign. It provided an overview of the research in spatial wayfinding and summarized aspects of business processes in public transportation. Finally it discussed the concept of epistemology and provided an overview of Frank's 5-tier ontology theory.

The conceptual model was built based on a reduced concrete case study. Chapter 4 elaborated on the conceptual model. At first, it discussed the processes and operations performed during a trip with public transportation. This chapter described the components of the conceptual model as the passenger, the point network, the vehicle, and the operations.

Chapter 5 presented an overview of the theory of data structures. State transition diagram and entity relationship theory were introduced. The chapter discussed the formation of algebras for representation of real world phenomena, and finally the functional programming languages. The above were used as tools to formally structure the conceptual scheme.

Chapter 6 examined the ontology of multi-modal public transportation. The chapter discussed the units of operation from the ontological point of view. It distinguished between the spatial wayfinding and the business process of a trip with public transportation. This chapter discussed the container operations and elaborated on the state transitions within a trip. It demonstrated the connection between information needs and the state transitions. Finally, it utilized and expanded Frank's 5-tier ontology scheme (Frank 2005-Draft) in public transportation.

Chapter 7 described the formalization of the state transition diagram and the conceptual point network. At first, it discussed the algebras of spatial wayfinding and business processes and the container algebras. Subsequently, it presented the components of the formal model for the state transition diagram. It also described the formalization of the conceptual point network of the reduced case.

Appendix I provides the Haskell encoding of the state transition diagram of a simplified trip. Appendix II is the Haskell encoding of the knowledge in the world for the reduced case.

8.2 Results and conclusions

The motivating scenarios and the reduced case showed that the operations performed during a trip with public transportation and the related information are hierarchical refinements of coarse, medium and fine levels of detail. The scenarios demonstrated that *there is a correspondence between the phase of a trip, the operation level and the level of detail in the information needed.*

This research considered the ontological decomposition into processes and operations as a tool for understanding the traveler's ability to perceive and interact with the transportation system. It described the ontology of the wayfinding task with public transportation as a composition of the "spatial wayfinding" and "passenger legitimation" processes. The processes were further structured into the units of "move", "contain", and "buy and sub-units" such as "enter", "exit", and "keep" operations found in the lower level.

The research formalized a passenger as a cognitive agent who transitions from one state to another based on his plans, perception, and strategy by performing operations. The formalization was introduced with the use of algebraic specifications of the operations. The agent's perception was modeled as affordances of the transportation network. The decision was modeled as the agent's strategy, namely the comparison of the potential next state and its associated afforded operation with the state stored in the mental plan. The actions were the operations performed by the agent and represented the outcome of the agent's strategy.

The agent's mental plan modeled the information needs in the form of planned information in the head. The state transition diagram modeled the knowledge embedded in the world. When there is not enough information to move between any two consecutive states, the overall task can not be accomplished. The information needed for making a complete trip is the collection of the information needed by the agent to move from any one state to the next. *Simulation with agent showed that the overall wayfinding task consists of a spatial wayfinding and business component.*

The identification of the agent's potential states involved answering a minimum set of questions at each point of the point network. *The answers to these questions are connected to the minimum information needs required by the agent to complete his trip legitimately with public transportation.* These are: need for ticket, ticket price, form of payment, ticket vendor location, need for ticket validation, ticket validation artifact location, vehicle stop location, vehicle line, vehicle destination, vehicle exit stop, and destination location.

8.3 Future research

The current research considers a non handicap adult with average abilities. Future research will benefit from the identification of information needs for special groups. These groups can include children, wheel chair or blind passengers and experience or inexperienced users of the public transportation system.

In the current research, the agent has a mental plan of his intended journey. The mental plan is not necessarily a correct plan. This formal implementation considers that the agent's plan is correct. The motivating scenarios demonstrated that some of the entries of the mental plan can be wrong. Future research will benefit from the consideration of incorrect mental plans. This research focus can be linked and examined in combination to risk. Another important aspect for future research is found in the case where the agent can not reach his spatial wayfinding goal or where he wishes to recover from mistakes.

Object and task granularity can yield results in special focus areas. This research can be expanded to further understand the different types of tickets. The analysis of the functionalities offered by two different types of ticket vending machines can assist in this direction.

The quotes from well known fiction inserted at the beginning of each chapter suggest that formal research can benefit from considering text in novels relevant to the task as the author looks very thoroughly into the scene he describes.

9. REFERENCES

Allen, G. L. (1997). From Knowledge to Word to Wayfinding: Issues in the Production and Comprehension of Route Directions. Spatial Information Theory - A Theoretical Basis for GIS (International Conference COSIT'97). S. C. Hirtle and A. U. Frank. Berlin-Heidelberg, Springer-Verlag. **1329:** 363-372.

Allen, G. L. (1999). Spatial Abilities Cognitive Maps and Wayfinding - Bases for Individual Differences in Spatial Cognition and Behavior. Wayfinding Behavior - Cognitive Maps and Other Spatial Processes. R. Golledge. Baltimore, Johns Hopkins University Press: 46-81.

Aristotelis Ethica Nicomachea. London, Oxford University Press Warehouse, Amen Corner, E.C. 1890.

Aristotle (1941 (reprint)). The Basic Works of Aristotle. New York, USA, Random House, Inc.

Beckett, S. (1959). Watt. New York, Grove Press.

Bittner, S. (2001). An agent-based model of reality in a cadastre. Department of Geoinformation. Vienna, Technical University Vienna.

Bond, A. and L. Gasser, Eds. (1988). Readings in Distributed Artificial Intelligence. San Mateo, Morgan Kaufmann.

Britannica (2003). Encyclopedia Britannica, Encyclopedia Britannica Corporation.

Chen, P. P.-S. (1976). "The entity-relationship model—toward a unified view of data." ACM Transactions on Database Systems (TODS) **1**(1, Special issue: papers from the international conference on very large data bases: September 22&ndash24, 1975, Framingham, MA): 9-36.

Chomsky, N. (1967). The Formal Nature of Language. The Biological Foundations of Language. E. H. Lenneberg. New York, Wiley and Sons: 397-442.

Chomsky, N. (1988). Language and Problems of Knowledge: The Managua Lectures. Cambridge, Massachusetts, The MIT Press.

Clarke, K. C. and L. J. Gaydos (1998). "Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for the San Francisco and Washington/Baltimore." International Journal of Geographical Information Science **12**(7): 699-714.

Codd, E. (1979). "Extending the Database Relational Model to Capture More Meaning." ACM TODS **4**(4): 379-434.Rel. Model Tasmania

Dastani, M. and L. van der Torre (2002). A Classification of Cognitive Agents. 24th Annual Meeting of the Cognitive Science Society (Cogsci'02), Washington DC, USA.

Denis, M., F. Pazzaglia, et al. (1999). "Spatial Discourse and Navigation: An Analysis of Route Directions in the City of Venice." Applied Cognitive Psychology **13**.

Dewey, J. (1938). Logic: The Theory of Inquiry. New York, Holt, Rinehart and Winston.

Dhar, R. (1997). "Consumer preference for a no-choice option." Journal of Consumer Research **24**: 215-231.

DiFebbraro, A., V. Recagno, et al. (1996). "INTANET: A New Simulation Tool for Intermodal Transportation System." Simulation Practice and Theory **4**: 47-64.

Dreyfus, H. L. (1981). From Micro-Worlds to Knowledge Representation: AI at an Impasse. Mind Design. J. Haugeland, Cambridge, Mass.: MIT Press: 161-204.

Eco, U. (1997). Kant and the platypus: essays on language and cognition. San Diego, CA, A Harvest Book Harcourt, Inc.

Edwards, W. (1967). The theory of decision making. Decision Making. W. Edwards and A. Tversky. USA, Penguin Books.

Einstein, A. (1934). Essays in Science, Barnes & Noble Books.

Ferber, J., Ed. (1998). Multi-Agent Systems - An Introduction to Distributed Artificial Intelligence, Addison-Wesley.

Frank, A., S. Bittner, et al. (2001). Spatial and Cognitive Simulation with Multi-agent Systems. Int. Conference COSIT 2001, Morro Bay, USA, September 2001, Lecture Notes in Computer Science, Vol. 2205, Berlin Heidelberg, Springer-Verlag.

Frank, A. U. (1992). Spational Reasoning - Theoretical Considerations and Practical Applications. EGIS '92 Third European Conference and Exhibition on Geographical Information Systems, München, Deutschland, EGIS Foundation.

Frank, A. U. (1999). One step up the abstraction ladder: Combining algebras - from functional pieces to a whole. Spatial Information Theory - Cognitive and Computational Foundations of Geographic Information Science (Int. Conference COSIT'99, Stade, Germany). C. Freksa and D. M. Mark. Berlin, Springer-Verlag. **1661:** 95-107.

Frank, A. U. (2001). An ontology in connected tiers for Geographic Information Systems. Zurich.

Frank, A. U. (2001). "Tiers of Ontology and Consistency Constraints in Geographic Information Systems." International Journal of Geographical Information Science **75**(5 (Special Issue on Ontology of Geographic Information)): 667-678.

Frank, A. U. (2003). Pragmatic Information Content: How to Measure the Information in a Route Description. Perspectives on Geographic Information Science. M. Goodchild, M. Duckham and M. Worboys. London, Taylor and Francis: 47-68.

Frank, A. U. (2003 Unpublished). Put verbs into the ontology! A formal ontology with processes. Institute for Geoinformation. Vienna, Technical University Vienna: 18.

Frank, A. U. (2005-Draft). Ontology for GIS. Vienna: 450.

Frank, A. U. (2005, Submitted). Practical Geometry—Mathematics for Geographic Information Systems.

Frank, A. U. and W. Kuhn (1995). Specifying Open GIS with Functional Languages. Advances in Spatial Databases (4th Int. Symposium on Large Spatial Databases, SSD'95, in Portland, USA). M. J. Egenhofer and J. R. Herring, Springer-Verlag. **951:** 184-195.

Freksa, C. (1992). Using Orientation Information for Qualitative Spatial Reasoning. Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. A. U. Frank, I. Campari and U. Formentini. Heidelberg-Berlin, Springer Verlag. **639:** 162-178.

Freksa, C. (1999). "Spatial Aspects of Task-Specific Wayfinding Maps."

Gaver, W. (1991). Technology affordances. CHI'91, New Orleans, Lousiana, USA, ACM, New York.

Gerbé, O. (1997). Conceptual Graphs for Corporate Knowledge Repositories. ICCS 1997. Fifth International Conference on Conceptual Structures, ICCS'97: Fulfilling Peirce's Dream, Seattle, Washington, USA, Springer.

Gibson, J. J. (1986). The Ecological Approach to Visual Perception. Hillsdale, NJ, Lawrence Erlbaum.

Golledge, R. (1999). Human Wayfinding and Cognitive Maps. Wayfinding Behavior - Cognitive Maps and Other Spatial Processes. R. Golledge. Baltimore, Johns Hopkins University Press: 5-45.

Golledge, R. (2005). Cognitive Maps. Encyclopedia of Social Measurement, Elsevier. **1:** 329-339.

Graham, W., R. Cassady, et al. (2000). "Modeling Intermodal Transportation Systems: Establishing a Common Language." National Center for Intermodal Transportation-DTRS98-G0017.

Gruber, T. R. (1993). "A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220, 1993."

Guarino, N. (1995). "Formal ontology, conceptual analysis and knowledge representation." International Journal of Human and Computer Studies. Special Issue on Formal Ontology, Conceptual Analysis and Knowledge Representation, edited by N. Guarino and R. Poli **43**(5/6).

Hall, E. T. (1966). The Hidden Dimension. Garden City, New York, Doubleday &

Company, Inc.

Haskell (1999). The Haskell 98 Report.

Heisenberg, W. (1983). Tradition in Science. San Francisco, California, Seabury Press.

Hochmair, H. (2000). Comparing cognitive maps for wayfinding tasks in the real world and the web (abstract). International Conference of Spatial Cognition, Rome, Italy, Pabst.

Hochmair, H. (2002). The Wayfinding Metaphor - Comparing the Semantics of Wayfinding in the Physical World and the WWW. Institute for Geoinformation Vienna. Vienna, Technical University Vienna.

Hofer, B. H. (2004). The Establishment of a Set of Tools for Map Use in a Digital Environment, Fachhochschule Technikum Kärnten. **Diploma:** Villach.

Johnson, M. (1987). The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason. Chicago, University of Chicago Press.

Kipling, R. (1901). Kim. Garden City, NY USA, Doubleday, Page & Company.

Kivel, R. P. (2006). Business Models. Do you have the right ingredients? The Nuts and Bolts of Business Plans.

Krek, A. (2002). An Agent-Based Model for Quantifying the Economic Value of Geographic Information. Vienna, University of Technology. **PhD**.

Kuhn, W. (2000). Ontologies from Texts. GIScience, Savannah, Georgia, USA.

Kuipers, B. (1978). "Modeling Spatial Knowledge." Cognitive Science **2**(2): 129-154.

Kuipers, B. (1982). "The 'Map in the Head' Metaphor." Environment and Behavior **14**(2): 202-220.

Lakoff, G. (1988). Cognitive Semantics. Meaning and Mental Representations. U. Eco, M. Santambrogio and P. Violi. Bloomington, Indiana University Press: 119-154.

Lakoff, G. and M. Johnson (1980). Metaphors We Live By. Chicago, University of Chicago Press.

Levy, S. (1993). Automatic Generation of Hyperbolic Tiling. The Visual Mind, Art and Mathematics. M. Emmer. Cambridge, Massachusetts, The MIT Press.

Liskov, B. and J. Guttag (1986). Abstraction and Specification in Program Development. Cambridge, Mass., The MIT Press.

Lott, J. S. (1993). "Intermodal Transit Station Computer Simulation Model." Automated People Movers **4**: 172-182.

Luce, R. D. (1967). Psychological Studies of Risky Decision Making. Decision Making. W. Edwards and A. Tversky. USA, Penguin Books.

Mark, D., C. Freksa, et al. (1999). "Cognitive Models of Geographical Space." IJGIS **13**(8): 747-774.

Mark, D., B. Smith, et al. (1999). Ontology and Geographic Objects: An Empirical Study of Cognitive Categorization. Spatial Information Theory. International Conference COSIT '99, Springer.

Matson, E. and S. DeLoach (2005). Formal Transition in Agent Organizations. International Conference Integration of Knowledge Intensive Multi-Agent Systems. KIMAS '05: Modeling, Evolution and Engineering, Waltham, Massachusetts, USA, IEEE Inc.

Minsky, M. (1985). The Society of Mind. New York, Simon & Schuster.

Morris, W., Ed. (1973). The American Heritage Dictionary of the English Language. Boston, New York, Atlanta, Geneva, Illinois, Dallas, Palo Alto, American Heritage Publishing Co., Inc. and Houghton Mifflin Company.

Nievergelt, J. and K. H. Hinrichs (1993). Algorithms & Data Structures: with Applications to Graphics and Geometry. Englewood Cliffs, New Jersey, Prentice-Hall.

Norman, D. A. (1988). The Design of Everyday Things. New York., Basic Books.

Osterwalder, A. (2004). The Business Model Ontology. A Proposition in a Design Science Approach. Ecole des Hautes Etudes Commercieles. Lausanne, Universite de Lausanne. **Docteur en Informatique de Gestion.**

Pass, C., B. Lowes, et al. (2000). Economics. Glasgow, Harper Collins Publishers.

Payne, J. W., J. R. Bettman, et al. (1993). The Adaptive Decision Maker, Cambridge University Press.

Phillips, L. and W. Edwards (1967). Conservatism in a Simple Probability Inference Task. Decision Making. W. Edwards and A. Tversky. USA, Penguin Books.

Piaget, J. and B. Inhelder (1967). The Child's Conception of Space. New York, Norton.

Pierce, C. S. (1992). The Essential Pierce. Selected Philosophical Writings, Volume II (1893-1913). Bloomington, Indiana USA, Indiana University Press.

Plato (370 B.C. (reprint 1953)). The Dialogues of Plato, Oxford University Press.

Pontikakis, E. (2004). Formal Encoding of Multi-Modal Trip with the Use of Public Transport - A Passenger's Perspective. AGILE 2004 7th Conference on Geographic Information Science, Heraklion, Crete, Greece, Crete University Press.

Pontikakis, E. (2005). *Contribution of Local Artifacts in Assessing Spatial Experiences What You Keep, What You Throw*. CORP 2005 & Geomultimedia05, Vienna, Austria, Selbstverlag des Institutes für EDV-gestützte Methoden in Architektur und Raumplanung.

Priss, U. (2001). *A Triadic Model of Information Flow. Conceptual Structures: Extracting and Representing Semantics*. 9th International Conference on Conceptual Structures, Stanford University, California, USA, Mineau.

Raubal, M. (2001). Agent-based simulation of human wayfinding -A model for unfamiliar buildings. *Department for Geoinformation*. Vienna, Technical University Vienna.

Rottenbacher, C., C. Achatschitz, et al. (2005, In preparation). *Treasure Hunt, Observation in Spatial Problem Solving*. Geospatial Semantics, Mexico City, Mexico.

Russell, B. (1956). *Portraits from Memory: And Other Essays*. New York, Simon and Schuster.

Russell, S. J. and P. Norvig (1995). *Artificial Intelligence-A Modern Approach*. Englewood Cliffs, NJ, Prentice Hall.

Schröder, E. (1890). *Vorlesungen über die Algebra der Logik (Exakte Logik)*. Leipzig, Teubner.

Searle, J. (1995). *The Construction of Social Reality*. New York, The Free Press.

Sivashanmugam, K. "Functional Languages and their uses in Web Services."

Smith, A. (1986 (original 1776)). *An Inquiry into the Nature and Causes of the Wealth of Nations*. New York.

Tanimoto, S. L. (1987). *The Elements of Artificial Intelligence*, Computer Science Press.

Thompson, S. (1996). *Haskell - The Craft of Functional Programming*. Harlow, UK, Addison-Wesley.

Timpf, S. (2002). "Ontologies of Wayfinding: a Traveler's Perspective." *Networks and Spatial Economics*(2): 9-33.

Timpf, S. and A. U. Frank (1997). Using Hierarchical Spatial Data Structures for Hierarchical Spatial Reasoning. *Spatial Information Theory - A Theoretical Basis for GIS (International Conference COSIT'97)*. S. C. Hirtle and A. U. Frank. Berlin, Springer-Verlag. **Lecture Notes in Computer Science 1329:** 69-83.

Timpf, S., G. S. Volta, et al. (1992). A Conceptual Model of Wayfinding Using Multiple Levels of Abstractions. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. A. U. Frank, I. Campari and U. Formentini. Heidelberg-Berlin, Springer Verlag. **639:** 348-367.

Tversky, B. and U. P. Lee (1999). Pictorial and Verbal Tools for Conveying Routes. Spatial Information Theory (Int. Conference COSIT'99, Stade, Germany), Springer-Verlag.

Uschold, M. and M. Grüninger (1996). "Ontologies: Principles, Methods and Applications." Knowledge Engineering Review **II**(2).

Werthner, H. (2003). Intelligent Systems in Travel and Tourism. 18th International Joint Conference on Artificial Intelligence, IJCAI-03, Acapulco, Mexico.

Whorf, B. L. (1956). Language, Thought, and Reality. Selected writings of Benjamin Lee Whorf. Cambridge, Massachusetts, The M.I.T. Press.

Winter, S. and S. Nittel (2003). "Formal Information Modeling for Standardisation in the Spatial Domain." International Journal of Geographical Information Science **17**(8): 721-741.

Woo, H. K. H. (1992). Cognition Value and Price, The University of Michigan Press.

WordNet (2.0). B. Gustafson, Princeton University Cognitive Science Lab.

Wurman, R. S. (1996). Information Architects. Zurich, Switzerland, Graphics Press Corporation.

```
module Passenger where
import List
import PointNet
import Ticket


--The StateAID is one of the important types.
--FairPaid and InVehicle are not used at the moment, but it helps describe the state
transitions
data StateA = StateA StateAID AtPOI InVehicle FairPaid deriving (Show, Eq)
type StateAID  = Int
type FairPaid  = Bool
type InVehicle = Bool


data Agent = Agent StateAID MentalPlan Pocket deriving (Show, Eq)
type MentalPlan = [StateAID]
type Pocket = Int

data OrigStAID = OrigStAID StateAID
data DestStAID = DestStAID StateAID

--
--
testPocket :: Pocket -> TicketPrice -> Bool
testPocket pockt tckPrc =
       if (pockt >= tckPrc) then True
       else False


------------------------------------------------------------------------------


module PointNet where
import List

data PntsOfIntst = PntsOfIntst AtPOI POIID
type AtPOI      = POIID
type POIID      = Int


------------------------------------------------------------------------------


module Ticket where
import List

data Ticket = Ticket TicketPrice Validated deriving (Show, Eq)
type TicketPrice       = Int
type Validated         = Bool


------------------------------------------------------------------------------


module Vehicle where
import List
import PointNet

-- State of Vehicle data structure
data Vehicle = Vehicle StateV

data StateV = StateV StateVID AtPOI Move deriving (Show)
type StateVID  = Int


data Move = CW | CCW | STOP deriving (Show)
-- I possibly only need the one for one way
-- CW stands for clockwise and CCW for counterclockwise
```

```
module StateTransitionWorld where
import List
import Vehicle
import PointNet
import Passenger
import Ticket


data StateTransitionWorld = StateTransitionWorld Agent Vehicle StateATransChart

type StateATransChart = [StateATrans]
type StateATrans = (StateAID, PotentStAIDOper)
type PotentStAIDOper = [StAIDOp]
type StAIDOp = (StateAID, Operation)

data Operation = DoNothing | DoWalk | DoLoseTick | DoPay |DoValidate
        |DoBoard | DoExit | DoStayInVeh | NullOper deriving (Show, Eq)
--

class FindNext stateAID where
        findNextMPlan :: stateAID -> MentalPlan ->stateAID
        findLineInChart :: stateAID -> StateATransChart -> PotentStAIDOper

instance FindNext StateAID where
--                                              Run with: findNextMPlan 3 mentalPlan2 ::
StateAID
        findNextMPlan st [] = 0
        findNextMPlan st (mPelem:mPelems) = if (st == mPelem) then head mPelems
                else findNextMPlan st mPelems

        findLineInChart a [] = error "Empty StateATransChart"
        findLineInChart a ((stAID,linePotSt):lines) = if a == stAID then linePotSt
                else findLineInChart a lines
--
findNxtLnInChart :: StateAID -> MentalPlan -> StateATransChart -> PotentStAIDOper
findNxtLnInChart st [] stAChr = error "Empty mentalMap or End reached"
findNxtLnInChart st (mPl:mPls) stAChr = findLineInChart (findNextMPlan st (mPl:mPls))
stAChr
--
--              Run with: findOper 7 [(6,DoNothing),(7,DoPay),(8,DoWalk)] :: Operation
class Operations operation where
        findOper :: StateAID -> PotentStAIDOper -> operation

instance Operations Operation where
        findOper a [] = error "NullOper"
        findOper a ((stAID,op):tl) = if a == stAID then op
                        else findOper a tl

findStAIdOper :: StateAID -> PotentStAIDOper -> StAIDOp
findStAIdOper a [] = error "No potential states"
findStAIdOper a ((stAID,op):tl) = if a == stAID then (stAID,op)
                        else findStAIdOper a tl
--
isStAIdIn :: StateAID -> PotentStAIDOper -> Bool
isStAIdIn a [] = False
isStAIdIn a ((stAID,op):tl) = if a == stAID then True
        else isStAIdIn a tl

-- At this point the passenger/Agent is ready to make the step that involves payment.

needToPay :: StateAID -> PotentStAIDOper -> Bool
needToPay st [] = error "No need to Pay"
needToPay st potIdOp =
        if ((findOper st potIdOp) == DoPay)
                then True
                else False
-- At this point the passenger/Agent is ready to make the step that involves validation.

needToVal :: StateAID -> PotentStAIDOper -> Bool
needToVal st [] = error "No need to Validate"
needToVal st potIdOp =
        if ((findOper st potIdOp) == DoValidate)
```

```
                then True
                else False

--

--                                      Run with: payFromPocket DoPay 9 6 :: Pocket
class Payments pocket where
        payFromPocket :: Operation -> pocket -> TicketPrice -> pocket

instance Payments Pocket where
        payFromPocket op pockt tckPrc=
                if (op == DoPay)
                        then if (testPocket pockt tckPrc == True) then pockt - tckPrc
                        else error "Not enough money in pocket"
                else pockt
--

--                                      Run with: validate DoValidate 9 6 :: Validated
class Validates ticket where
        validate :: Operation -> ticket -> ticket

instance Validates Ticket where
        validate op (Ticket pr val) =
                if (op == DoValidate)
                        then (Ticket pr True)
                else error "No need for validation"

--

makeOneStep :: Agent -> TicketPrice -> StateATransChart -> StAIDOp
makeOneStep (Agent st [] pkt) tckPrc stAChr = error "Empty mentalMap or End reached"
makeOneStep (Agent st (mPl:mPls) 0) tckPrc stAChr = error "Unable to commute: Empty pocket"
makeOneStep (Agent st (mPl:mPls) pkt) tckPrc stAChr =
        if st == mPl
                        then if ((payFromPocket (snd (findStAIdOper (head mPls)
                         (findLineInChart st stAChr))) pkt tckPrc) >= 0)
                                then findStAIdOper (head mPls) (findLineInChart st stAChr)
                        else error "Not enough money for ticket in pocket"
                else makeOneStep (Agent st mPls pkt) tckPrc stAChr
--
make1StepNewAg :: Agent -> TicketPrice -> StateATransChart -> Agent
make1StepNewAg (Agent st [] pkt) tckPrc stAChr = error "Empty mentalMap or End reached"
make1StepNewAg (Agent st (mPl:mPls) 0) tckPrc stAChr = error "Unable to commute: Empty
pocket"
make1StepNewAg (Agent st (mPl:mPls) pkt) tckPrc stAChr =
        if st == mPl
                        then if ((payFromPocket (snd (findStAIdOper (head mPls)
                         (findLineInChart st stAChr))) pkt tckPrc) >= 0)
                                then (Agent (fst (findStAIdOper (head mPls) (findLineInChart
                                 st stAChr))) (mPl:mPls) pkt1)
                        else error "Not enough money for ticket in pocket"
                else make1StepNewAg (Agent st mPls pkt) tckPrc stAChr
                        where
                                pkt1 = payFromPocket (snd (findStAIdOper (head mPls)
                                 (findLineInChart st stAChr))) pkt tckPrc

--
makeTrip :: Agent -> TicketPrice -> StateATransChart -> [Agent]
makeTrip (Agent st [] pkt) tckPrc stAChr = error "Empty mentalMap or End reached"
makeTrip (Agent st (mPl:mPls) 0) tckPrc stAChr = error "Get some money: Empty pocket"
makeTrip (Agent st (mPl:mPls) pkt) tckPrc stAChr =
        if (st == (head.reverse) mPls)
                then error "End of trip"
                else [make1StepNewAg (Agent st (mPl:mPls) pkt) tckPrc stAChr] ++
                        makeTrip (make1StepNewAg (Agent st (mPl:mPls) pkt) tckPrc stAChr)
                                tckPrc stAChr
```

```
module Data where
import StateTransitionWorld
import List
import Vehicle
import PointNet
import Passenger

--the current module can be later be brocken into data for stateTrans and data for
mentalPlan
--DATA STATE TRANSITION

--Test data for the Agent's transition of states with associated operations
stateATrans1 = (1, [(1,DoNothing),(2,DoWalk)]) :: StateATrans
stateATrans2 = (2, [(1,DoWalk),(2,DoNothing),(3,DoPay)]) :: StateATrans
stateATrans3 = (3, [(1,DoWalk),(2,DoLoseTick),(3,DoNothing),(4,DoBoard)]) :: StateATrans
stateATrans4 = (4, [(2,DoExit),(5,DoStayInVeh)]) :: StateATrans
stateATrans5 = (5, [(6,DoExit)]) :: StateATrans
stateATrans6 = (6, [(6,DoNothing),(7,DoPay),(9,DoWalk)]) :: StateATrans
stateATrans7 = (7, [(6,DoLoseTick),(7,DoNothing),(8,DoBoard),(9,DoWalk)]) :: StateATrans
stateATrans8 = (8, [(6,DoExit), (10,DoStayInVeh)]) :: StateATrans
stateATrans9 = (9, [(9,DoNothing),(6,DoWalk)]) :: StateATrans
stateATrans10 = (10, [(2,DoExit)]) :: StateATrans

--The stateATransChart1 represents the collections of affordances that objects "emit"
--in the world

stateATransChart1 = [stateATrans1,stateATrans2,stateATrans3,stateATrans4,
 stateATrans5,stateATrans6,stateATrans7,stateATrans8,stateATrans9,stateATrans10] ::
  StateATransChart
--DATA MENTAL PLAN

mentalPlan1 = [1, 2, 3, 4, 5, 6, 9] :: MentalPlan
mentalPlan2 = [9, 6, 7, 8, 10, 2, 1] :: MentalPlan


----TEST FUNCTIONS-------------------------------------------------------------------
top1, top2, top3 :: Operation
top1 = findOper 7 [(6,DoNothing),(7,DoPay),(8,DoWalk)]
top2 = findOper 6 [(6,DoNothing),(7,DoPay),(8,DoWalk)]
top3 = findOper 8 [(6,DoExit), (10,DoStayInVeh)]

tpk1, tpk2, tpk3 :: Pocket
tpk1 = payFromPocket DoExit 10 9
tpk2 = payFromPocket DoPay 10 12
tpk3 = payFromPocket DoPay 9 6

sid1 = 1 :: StateAID
sid2 = 2 :: StateAID
sid3 = 3 :: StateAID
sid4 = 4 :: StateAID
sid5 = 5 :: StateAID
sid6 = 6 :: StateAID


fln1, fln2, fln3, fln4, fln5, fln6 :: PotentStAIDOper
fln1 = findLineInChart sid1 stateATransChart1
fln2 = findLineInChart sid2 stateATransChart1
fln3 = findLineInChart sid3 stateATransChart1
fln4 = findLineInChart sid4 stateATransChart1
fln5 = findLineInChart sid5 stateATransChart1
fln6 = findLineInChart sid6 stateATransChart1

fnxt1, fnxt2, fnxt3 :: StateAID
fnxt1 = findNextMPlan 6 mentalPlan2
fnxt2 = findNextMPlan 3 mentalPlan2
fnxt3 = findNextMPlan 3 mentalPlan1

ndpay1 = needToPay 7 [(6,DoNothing),(7,DoPay),(8,DoWalk)]
ndpay2 = needToPay 6 [(6,DoNothing),(7,DoPay),(8,DoWalk)]
ndpay3 = needToPay 7 [(6,DoExit), (10,DoStayInVeh)]
ndpay4 = needToPay 7 [(6,DoNothing),(7,DoPay),(8,DoWalk)]
```

```
module PointNetScheme (OODB (..),
                 module Value240,
                 module ValueEx240,
                 module OODB320,
                 module PointNetScheme       -- exports all names!
                    ) where

{- db to test the put verbs into the ontology
based on dbempty, but imports oodb320

this is only the OODB definition, loading is moved to PointNetLoad
containedIn a set of operations which cannot be defined independent of OODB

af aug2003 - put verbs into ontology
E.Pontikakis: modified entities to fit public transport 'knowledge in the world'
---------------------------------------------------------------------}

import qualified Prelude
import qualified Maybe
import Value240
import ValueEx240
import OODB320
-------------------------
data FixedType =
                BusStopT      |
                BusStpOrdT    |
                BusLnPntT     |
                BusStopPntT   |
                VehDelayPntT  |
                BusDestT      |
                BusLineT      |
                AffordanceT   |
                AffrdBundleT  |
                AffrdBPntT    |
                AffrdToBndlT  |
                BusStopLineT  |
                DelayT        |
                OrderT        |
                PointT        |
                VehIDT
                      deriving (Eq, Ord, Show)
---------------------------------------------------------------------------
--data VehicleState = Mooving | Stopped deriving (Show, Eq)
--should go to different database group for the vehicle
---------------------------------------------------------------------------
type BusStop = String
type BusLine = String
type BusDest = String
type Affordance = String
type AffrdBundle = String
type AffrdToBndl = String                   --Affordance
type POIID = Int
type Order = Int
type Delay = Int
type BusStpOrd = Int                        --Order
type BusLnPnt = Int                         --POIID
type BusStopPnt = Int                       --POIID
type VehDelayPnt = Int                      --POIID
type AffrdBPnt = Int                        --POIID
type DelayBStpNB = Int                      --Delay
type VehID = Int
type VehIDLn = Int                          --VehID

data OODB = OODB {lastid::ID,
            name :: RelVal Name,
            busStop :: RelID,
            busLine :: RelID,
            busDest :: RelID,
            affordance :: RelID,
            affrdBundle :: RelID,
            affrdToBndl :: RelID,
```

```
            point :: RelVal POIID,
            order :: RelVal Order,
           delay :: RelVal Delay,
            busStpOrd :: RelID,
             busLnPnt :: RelID,
             busStopPnt :: RelID,
             vehDelayPnt :: RelID,
             affrdBPnt :: RelID,
             delayBStpNB :: RelID,
             vehID :: RelVal VehID,
             vehIDLn :: RelID,
            fixedType :: RelVal FixedType}                 --   deriving (Show)

instance Zeros OODB where
    zero = OODB zero zero zero zero zero zero zero zero zero zero zero zero
     zero zero zero zero zero zero zero zero
-- name busStop busLine busDest affordance affrdBundle affrdToBndl point order
-- delay busStpOrd busLnPnt affrdBPnt delayBStpNB vehID vehIDLn fixed type
------------------------------------------------------------- updates on the properties
doToName op db = db {name = op $ name db}
doToBusStop op db = db {busStop = op $ busStop db}
doToBusLine op db = db {busLine = op $ busLine db}
doToBusDest op db = db {busDest = op $ busDest db}
doToAffordance op db = db {affordance = op $ affordance db}
doToAffrdBundle op db = db {affrdBundle = op $ affrdBundle db}
doToAffrdToBndl op db = db {affrdToBndl = op $ affrdToBndl db}
doToPoint op db = db {point = op $ point db}
doToOrder op db = db {order = op $ order db}
doToDelay op db = db {delay = op $ delay db}
doToBusStpOrd op db = db {busStpOrd = op $ busStpOrd db}
doToBusLnPnt op db = db {busLnPnt = op $ busLnPnt db}
doToAffrdBPnt op db = db {affrdBPnt = op $ affrdBPnt db}
doToDelayBStpNB op db = db {delayBStpNB = op $ delayBStpNB db}
doToVehID op db = db {vehID = op $ vehID db}
doToVehIDLn op db = db {vehIDLn = op $ vehIDLn db}
doToPtBusStopName op db = db {busStopPnt = op $ busStopPnt db}
doToPtBusStopDelay op db = db {vehDelayPnt = op $ vehDelayPnt db}
------------------------------------------------------------- updates on the properties

deleteEntity i db = doToFixedType (removeIded i) db
        -- only the fixed type is removed, if type is checked, this would be sufficient...
          where
                doToFixedType op db = db {fixedType = op $ fixedType db}

-------------------------------------------------------------------------------------
--                                        generic stuff (should form its own module
--                                                  with import from PointNetScheme
class MgtIDs env i t where                                        -- t is the type
    insertNew :: t -> env -> (i, env)

instance MgtIDs OODB ID FixedType where
    insertNew t env = (i, env' {fixedType = (insertIdedUnchecked i t) $ fixedType env'})
         where   (i, env') = nextID env
-------------------------------------
class DBop b where
    insVal :: ((AssList ID b -> AssList ID b) -> c -> c) -> ID ->  b -> c -> c
instance Eq a => DBop a where
    insVal op i val oo = op (insertIdedUnchecked i val) oo

to' :: FromTos a b => [Char] -> ID -> (a -> RelVal b) -> a -> b
to' e i p db = fromJustGuarded (e ++^ show i ) . unique. to db p . singleton $ i
--                  why not generalizable for db
        -- because when putting a concrete attribute in, which is of type OODB -> AssList
         -- then the infered type is not general enough

identify :: OODB -> Name -> ID
identify db identifier = fromJustGuarded ("identify: finding id by identifying property "
                      ++ identifier) .unique. from db name . singleton $ identifier
--------------------------------------------------------------------------
instance IdCols ID OODB where
    lastID  = lastid
    setNewID i oo = oo {lastid = i}
```

```
------------------------------------------------------------------- EntityType Checking

findType :: ID -> OODB ->  [FixedType]  -- every id must have at least one entity type
findType i g = if null x then error ("findType - no type for " ++ show i)
                     else x
         where x = to g fixedType . singleton $ i
        -- maybe2error ("findType - no type for " ++ show i).--findVal unType VType g $ i

checkType :: FixedType -> ID -> OODB -> Bool
checkType et i db = et `e` (findType i db)

checkTypeM :: FixedType -> OODB -> Maybe ID-> Maybe ID
checkTypeM e db (Just i)  = if checkType e i db then Just i else Nothing
checkTypeM _ _ _ = Nothing

eTestID :: FixedType -> OODB -> ID -> ID
eTestID e db i = if checkType e i db then i
               else error ("iTest - " ++ show i ++ "does not have type " ++ show e ++ " but
"                     ++ show (findType i db) )

eTestDB :: FixedType -> ID -> OODB -> OODB
eTestDB e i db = if checkType e i db then db
               else error ("iTest - " ++ show i ++ "does not have type " ++ show e ++ " but
"                     ++ show (findType i db) )
```

```
module PointNetLoad (module PointNetLoad,
                module PointNetScheme)  where


{-af aug2003 - put verbs into ontology
E.Pontikakis: modified entities to fit public transport data entities -}

import qualified Prelude (writeFile)
import PointNetScheme
import List (zipWith4, zipWith5)
type DB = OODB
------------------------------------- loading the database
o0, o1 :: OODB
o0 = zero
--o1 = foldr storeBusDest o0 busDestData
o1 = foldr storeBusLine o0 busLineData
o1b = foldr storeOrder o1 orderData

o2 = foldr storeBusStop o1b busStopData
o3 = foldr storeAffordance o2 affordanceData
o4 = foldr storeAffrdBundle o3 affrdBundleData
o5 = foldr storePoint o4 pointData
o6 = foldr storeDelay o5 delayData
o7 = foldr storeVehID o6 vehIDData

o2a = foldr storeBusDestLineRel o7 busLineDestData
o2b = foldr storeBusStopOrdRel o2a busStopOrdRelData
o2c = foldr storeBusStopLineRel o2b busLineStopData
o2d = foldr storeBusLinePntRel o2c busLnPntRelData
o2e = foldr storeAffrdBPntRel o2d affrdBPntRelData
o2f = foldr storeAffrdToBndlRel o2e affrdToBndlRelData
o2g = foldr storeVehIDLnRel o2f vehIDLnData
o2h = foldr storePtBusStopNameRel o2g pntBusStopNameData
o2i = foldr storePtBusStopDelayRel o2h ptBusStopDelayData


oN = o2i      -- oN is exported!
-- To write the oN dataBase in a file,
-- execute the wrtFl. The default directory is the desktop
wrtFl = Prelude.writeFile "oNData" (show oN)
----------------------------------------------
-- convert one assList to text with header

vlist st att db =  (st :) . map show . unasslist. att $ db
ilist st att db =  (st :) . map show. filter (notUnknown.snd) . unasslist. att $ db
-- vlist for lists with values, ilist for relations
class Unk a where
    notUnknown :: a -> Bool
instance Unk ID where
    notUnknown (ID a) = True
    notUnknown (IDunknown) = False

toLength :: a -> Int -> [a] -> [a]
toLength z n st = if l >0 then  st ++ (Prelude.replicate l z ) else Prelude.take n st
    where l = n - Prelude.length st

mapop :: [(a->b)] -> a ->  [b]
mapop [] x = []
mapop (op:ops) x = (op x) : (mapop ops x)
----------------------------
merge4 :: Zeros a => [a] -> [(a,a,a,a)]
merge4 [] = []
merge4 (a:[]) = [(a,zero,zero,zero)]
merge4 (a:b:[]) = [(a,b,zero,zero)]
merge4 (a:b:c:[]) = [(a,b,c,zero)]
merge4 (a:b:c:d:[]) = [ (a,b,c,d) ]
merge4 (a:b:c:d:rest) = (a,b,c,d):  merge4 rest
--------------
merge5 :: Zeros a => [a] -> [(a,a,a,a,a)]
merge5 [] = []
merge5 (a:[]) = [(a,zero,zero,zero, zero)]
merge5 (a:b:[]) = [(a,b,zero,zero, zero)]
```

```
merge5 (a:b:c:[]) = [(a,b,c,zero, zero)]
merge5 (a:b:c:d:[]) = [ (a,b,c,d,zero) ]
merge5 (a:b:c:d: e: rest) = (a,b,c,d, e):  merge5 rest
------------------
makeLine :: Int -> ([String], [String], [String], [String]) -> [String]
makeLine n (a,b,c,d) = zipWith4 op1 (toLength zs l a) (toLength zs l b) (toLength zs l c)
(toLength zs l d)
                        ++ ["\n"]
    where   l = max (max (Prelude.length a) (Prelude.length b)) ( Prelude.length c)
            zs = Prelude.replicate n ' '
            op = toLength ' ' n
            op1  a b c d = op a ++ op b ++ op c ++ op d ++ "\n"
----------------
makeLine5 n (a,b,c,d, e) = zipWith5 op1 (toLength zs l a) (toLength zs l b)
            (toLength zs l c) (toLength zs l d) (toLength zs l e)
                        ++ ["\n"]
    where   l = max ( Prelude.length e) (max (max (Prelude.length a) (Prelude.length b)) (
Prelude.length c))
            zs = Prelude.replicate n ' '
            op = toLength ' ' n
            op1  a b c d e = op a ++ op b ++ op c ++ op d ++ op e ++ "\n"
-----------------

instance Show OODB where
    show a = -- concat . mergetolines . someList $ a
            concat. concat $ (map (makeLine 22) . merge4 . vList $ a)
                         ++ (map (makeLine5 20) . merge5 . iList $ a )

iList :: OODB -> [[String]]
iList db = mapop [ilist "BusStop:BusLine" busStop,
 ilist "BusDest:BusLine" busDest, ilist "BusStpOrd:BusStop " busStpOrd,
  ilist "BusLnPnt:BusLine" busLnPnt, ilist "AfrdBPnt:AfrdBndl" affrdBPnt,ilist
       "Pnt:BusStopName" busStopPnt, ilist "Afrd:AfrdBndl" affrdToBndl, ilist "VehID:BusLn"
              vehIDLn, ilist "Pnt:DelayNB" vehDelayPnt] db
vList db = mapop [vlist "POIID" point,vlist "Name" name, vlist "Order" order, vlist "Delay"
       delay, vlist "VehID" vehID,vlist "FixedType" fixedType] db
------------------------------------------------------------------------------------
storeBusLine (n) db = db2
       where
              (i, db1) = insertNew BusLineT db
              db2 = insVal doToName i n db1

busLineData = [("37N"),("38N"),("37S"),("38S")] :: [(String)]
------------------------------------------------------------------------------------
{-
storeBusDest (n) db = db2
       where
              (i, db1) = insertNew BusDestT db
              db2 = insVal doToName i n db1

busDestData = [("Hohe Warte"), ("Grinzing"), ("Schottentor")] :: [(Name)]
-}
------------------------------------------------------------------------------------
storeBusStop (n) db = db2
       where
              (i, db1) = insertNew BusStopT db
              db2 = insVal doToName i n db1

busStopData =
[("NoLabel"),("Schottentor"),("NussdorferS"),("GlatzG"),("HardtG"),("GatterburgG"),
       ("SilberG"),("Oberdoebling"),("Grinzing"),("GuneschG"),
        ("DoeblHGatterburgG"),("Hohe Warte")] :: [(BusStop)]


--Run:  to oN name [ID 33]            Returns: ["HardtG"]
-----------------------------------------------------------------------------------
storeOrder (n) db = db2
       where
              (i, db1) = insertNew OrderT db
              db2 = insVal doToOrder i n db1

orderData = [0..20] :: [Int]
```

```
--------------------------------------------------------------------------------
storeVehID (n) db = db2
       where
               (i, db1) = insertNew VehIDT db
               db2 = insVal doToVehID i n db1

vehIDData = [100..499] :: [Int]
--------------------------------------------------------------------------------

{-
from oN busStpOrd . singleton $ 2      :result [# 18]
to oN busStpOrd . singleton . ID $ 12        :result [8]
-}
--------------------------------------------------------------------------------
storeBusStopOrdRel :: (BusStpOrd, BusStop) -> OODB -> OODB
storeBusStopOrdRel (bstpOrd, bstp) db = db1
       where
               bstpOrd' = head (from db order . singleton $ bstpOrd)
               bstp' = identify db bstp

               db1 = insVal doToBusStpOrd bstpOrd' bstp' db

busStopOrdRelData = [(0,"NoLabel"),(1,"Schottentor"),(2,"NussdorferS"),(3,"GlatzG"),
 (4,"HardtG"),(5,"GatterburgG"),(6,"SilberG"),(7,"Oberdoebling"),(8,"Grinzing"),
  (9,"GuneschG"),(10,"DoeblHGatterburgG"),(11,"Hohe Warte")] :: [(Order, BusStop)]
--------------------------------------------------------------------------------
--NB northbound
storeDelayBStpNB :: (DelayBStpNB, BusStop) -> OODB -> OODB
storeDelayBStpNB (dlBstp, bstp) db = db1
       where
               dlBstp' = head (from db order . singleton $ dlBstp)
               bstp' = identify db bstp

               db1 = insVal doToDelayBStpNB dlBstp' bstp' db

busStopDelayNBData = [(0,"NoLabel"),(0,"Schottentor"),(2,"NussdorferS"),(1,"GlatzG"),
 (2,"HardtG"),(1,"GatterburgG"),(2,"SilberG"),(1,"Oberdoebling"),(1,"Grinzing"),
  (2,"GuneschG"),(1,"DoeblHGatterburgG"),(1,"Hohe Warte")] :: [(Delay, BusStop)]

--findLineByDest num = to oN busDest . from oN name . singleton $ num

--------------------------------------------------------------------------------

storePoint (n) db = db2
       where
               (i, db1) = insertNew PointT db
               db2 = insVal doToPoint i n db1

pointData = [1010,1020,1030,1040,1050,1060,1070,1080,1090,1101,1102,1103,
   1104,1110,1121,1122,1123,1124,
       1130,1140,1150,1160,1171,1172,1173,1174,1180,1192,1193,1200,1210,1220,1230,1242,
        1243,1250,1260,1270,1280,1292,1293,1300,1310,1320,1332,1333,1340,1350,1360,1370,
          1400,1410,1420,1430,1442,1443,1450,1460,1471,1474,1480,1490,1500,1511,1514,1520,
           1530,1541,1544] :: [(POIID)]
--------------------------------------------------------------------------------

storeBusLinePntRel :: (BusLnPnt, BusLine) -> OODB -> OODB
storeBusLinePntRel (bLnPnt, bln) db = db1
       where
               bLnPnt' = head (from db point . singleton $ bLnPnt)
               bln' = identify db bln
               db1 = insVal doToBusLnPnt bLnPnt' bln' db

busLnPntRelData = [(1121,"37N"),(1122,"38N"),(1123,"38S"),(1124,"37S"),
 (1101,"37N"),(1102,"38N"),(1103,"38S"),(1104,"37S"),
  (1171,"37N"),(1172,"38N"),(1173,"38S"),(1174,"37S"),
   (1192,"38N"),(1193,"38S"),(1242,"38N"),(1243,"38S"),(1292,"38N"),(1293,"38S"),
    (1332,"38N"),(1333,"38S"),(1442,"38N"),(1443,"38S"),
     (1471,"37N"),(1474,"37S"),(1511,"37N"),(1514,"37S"),(1541,"37N"),(1544,"37S")]
       :: [(POIID, BusLine)]
--------------------------------------------------------------------------------
storeAffrdBPntRel :: (AffrdBPnt, AffrdBundle) -> OODB -> OODB
```

```
storeAffrdBPntRel (afBPnt, afBdl) db = dbl
        where
                afBPnt' = head (from db point . singleton $ afBPnt)
                afBdl' = identify db afBdl
                dbl = insVal doToAffrdBPnt afBPnt' afBdl' db

affrdBPntRelData = [(1010,"AtNoInterest"),(1020,"AtHome"),(1030,"AtIntersect"),
  (1040,"AtIntersect"),(1050,"AtIntersect"),(1060,"AtIntersect"),(1070,"AtIntersect"),
   (1080,"AtIntersect"),(1090,"AtIntersect"),(1101,"AtBusStop"),(1102,"AtBusStop"),
    (1103,"AtBusStop"),(1104,"AtBusStop"),(1110,"AtIntersect"),(1121,"AtBusStop"),
     (1122,"AtBusStop"),(1123,"AtBusStop"),(1124,"AtBusStop"),(1130,"AtNoInterest"),
      (1140,"AtNoInterest"),(1150,"AtNoInterest"),(1160,"AtIntersect"),(1171,"AtBusStop"),
       (1172,"AtBusStop"),(1173,"AtBusStop"),(1174,"AtBusStop"),(1180,"AtIntersect"),
        (1192,"AtBusStop"),(1193,"AtBusStop"),(1200,"AtIntersect"),(1210,"AtIntersect"),
         (1220,"AtIntersect"),(1230,"AtIntersect"),(1242,"AtBusStop"),(1243,"AtBusStop"),
          (1250,"AtIntersect"),(1260,"AtIntersect"),(1270,"AtIntersect"),
           (1280,"AtIntersect"),(1292,"AtBusStop"),(1293,"AtBusStop"),
            (1300,"AtIntersect"),(1310,"AtIntersect"),(1320,"AtIntersect"),
             (1332,"AtBusStop"),(1333,"AtBusStop"),(1340,"AtIntersect"),
              (1350,"AtIntersect"),(1360,"AtIntersect"),(1370,"AtDest"),
               (1400,"AtIntersect"),(1410,"AtNoInterest"),(1420,"AtIntersect"),
                (1430,"AtIntersect"),(1442,"AtBusStop"),(1443,"AtBusStop"),
                 (1450,"AtNoInterest"),(1460,"AtIntersect"),(1471,"AtBusStop"),
                  (1474,"AtBusStop"),(1480,"AtIntersect"),(1490,"AtIntersect"),
                   (1500,"AtIntersect"),(1511,"AtBusStop"),(1514,"AtBusStop"),
                    (1520,"AtIntersect"),(1530,"AtIntersect"),(1541,"AtBusStop"),
                     (1544,"AtBusStop")] :: [(POIID, AffrdBundle)]
--storeAffrdBPntRel affrdBPntRelData

--ilist "Pnt:BusStopName" busLnPnt,
-----------------------------------------------------------------------------------------
storeAffrdToBndlRel :: (AffrdToBndl, AffrdBundle) -> OODB -> OODB
storeAffrdToBndlRel (afToBnd, afBdl) db = dbl
        where
                afToBnd' = identify db afToBnd
                afBdl' = identify db afBdl
                dbl = insVal doToAffrdToBndl afToBnd' afBdl' db

affrdToBndlRelData =
[("Wait","AtBusStop"),("PercvBus","AtBusStop"),("BuyTick","AtBusStop"),
("Validate","AtBusStop"),("Board","AtBusStop"),("Exit","AtBusStop"),
  ("Walk","AtBusStop"),("PercvBusStop","AtIntersect"),("Walk","AtIntersect"),
   ("Wait","AtIntersect"),("Walk","AtNoInterest"),("Wait","AtNoInterest"),
    ("Walk","AtHome"),("Wait","AtHome"),("Walk","AtDest"),
     ("Wait","AtDest")] :: [(Affordance, AffrdBundle)]

-----------------------------------------------------------------------------------------
--storeBusLinePntRel  busLnPntRelData
--reverse order in name of function
storeBusDestLineRel :: (String,String) -> OODB -> OODB
storeBusDestLineRel (bln, bdn) db = dbl
        where
                bdn' = identify db bdn
                bln' = identify db bln
                dbl = insVal doToBusDest bdn' bln' db
busLineDestData = [("38N","Grinzing"), ("37N","Hohe Warte"), ("38S","Schottentor"),
("37S","Schottentor")] :: [(String,String)]

findLineByDest num = to oN busDest . from oN name . singleton $ num
--run with: findLineByDest "Grinzing"
-----------------------------------------------------------------------------------------
storeBusStopLineRel :: (String,String) -> OODB -> OODB
storeBusStopLineRel (bln, bstp) db = dbl
        where
                bstp' = identify db bstp
                bln' = identify db bln
                dbl = insVal doToBusStop bstp' bln' db
busLineStopData =
[("38N","Schottentor"),("38S","Schottentor"),("37N","Schottentor"),("37S","Schottentor"),
 ("38N","NussdorferS"),("38S","NussdorferS"),("37N","NussdorferS"),("37S","NussdorferS"),
  ("38N","GlatzG"),("38S","GlatzG"),("37N","GlatzG"),("37S","GlatzG"),
   ("38N","HardtG"),("38S","HardtG"),("38N","GatterburgG"),("38S","GatterburgG"),
```

```
    ("38N","Grinzing"),("38S","Grinzing"),("38N","SilberG"),("38S","SilberG"),
     ("38N","Oberdoebling"),("38S","Oberdoebling"),("37N","GuneschG"),("37S","GuneschG"),
      ("37N","DoeblHGatterburgG"),("37S","DoeblHGatterburgG"),("37N","Hohe Warte"),
("37S","Hohe Warte")]
          :: [(String,String)]

findLineByStop num = to oN busStop . from oN name . singleton $ num

findBusStopsFromLine ln = from oN busStop. singleton $ identify oN ln
findBusStopsFromLineID lnID = from oN busStop. singleton . ID $ lnID

-------------------------------------------------------------------- - - - - - - - - - -
storePtBusStopNameRel :: (POIID,String) -> OODB -> OODB
storePtBusStopNameRel (pnt, busStpName) db = db1
       where
              pnt' = head (from db point . singleton $ pnt)
              busStpName' = identify db busStpName
              db1 = insVal doToPtBusStopName pnt' busStpName' db

pntBusStopNameData =
[(1101,"NussdorferS"),(1102,"NussdorferS"),(1103,"NussdorferS"),(1104,"NussdorferS"),
   (1121,"Schottentor"),(1122,"Schottentor"),(1123,"Schottentor"),(1124,"Schottentor"),
    (1171,"GlatzG"),(1172,"GlatzG"),(1173,"GlatzG"),(1174,"GlatzG"),
      (1192,"HardtG"),(1193,"HardtG"), (1242,"GatterburgG"),(1243,"GatterburgG"),
        (1292,"SilberG"),(1293,"SilberG"),(1332,"Oberdoebling"),(1333,"Oberdoebling"),
         (1442,"Grinzing"),(1443,"Grinzing"),(1471,"GuneschG"),
          (1474,"GuneschG"),(1511,"DoeblHGatterburgG"),(1514,"DoeblHGatterburgG"),
           (1541,"Hohe Warte"),(1544,"Hohe Warte")] :: [(POIID, String)]

findPointIDsFromBusStopsNameID busStName = from oN busStopPnt . singleton . ID $ busStName
--run with:  findPointIDsFromBusStopsNameID  35     --output: [#110,#109,#108,#107]
findPointIDsFromBusStopsName busStName = from oN busStopPnt . singleton $ identify oN
busStName
--run with:  findPointIDsFromBusStopsName "Grinzing"   --output: [#65,#64]

----------------------------------------------------------------------
storePtBusStopDelayRel :: (POIID,DelayBStpNB) -> OODB -> OODB
storePtBusStopDelayRel (pnt, dlay) db = db1
       where
              pnt' = head (from db point . singleton $ pnt)
              dlay' =  head (from oN delay . singleton $ dlay)
              db1 = insVal doToPtBusStopDelay pnt' dlay' db

ptBusStopDelayData = [(1101,2),(1102,2),(1121,0),(1122,0),(1171,3),(1172,3),
 (1192,5),(1242,6),(1292,7),(1332,8),(1442,9),(1471,5),(1511,6),(1541,7)]
  :: [(POIID, DelayBStpNB)]


--to oN vehDelayPnt . singleton . ID $ 110
--[#128]

--     from oN busStpOrd . singleton $ 2      :result [# 18]
--to oN busStpOrd . singleton . ID $ 12       :result [8]
-------------------------------------------------------------------------------------
--test busStop = identify  oN busStop

findNameFromID int = to oN name . singleton . ID $ int
--run with: findNameFromID 4
findIDFromID int = to oN busStop . singleton . ID $ int
--run with findIDFromID 12  returns [#6,#4]

-------------------------------------------------------------------------------------
storeAffordance (n) db = db2
       where
              (i, db1) = insertNew AffordanceT db
              db2 = insVal doToName i n db1

affordanceData = [("Wait"),("Walk"),("PercvBus"),("PercvBusStop"),("BuyTick"),
        ("Validate"),("Board"),("Exit")] :: [(Affordance)]

-------------------------------------------------------------------------------------
```

```
storeAffrdBundle (n) db = db2
        where
                (i, db1) = insertNew AffrdBundleT db
                db2 = insVal doToName i n db1

affrdBundleData = [("AtNoInterest"),("AtHome"),("AtIntersect"),("AtBusStop"),("AtDest")] ::
[(AffrdBundle)]
------------------------------------------------------------------------------------------
storeDelay (n) db = db2
        where
                (i, db1) = insertNew DelayT db
                db2 = insVal doToDelay i n db1

delayData = [0..10] :: [(Delay)]
-------------------------------------------------------------------- - - - - - - - - - - -


storeVehIDLnRel :: (Int,String) -> OODB -> OODB
storeVehIDLnRel (vhid, bsln) db = db1
        where
                vhid' = head (from db vehID . singleton $ vhid)
                bsln' = identify db bsln
                db1 = insVal doToVehIDLn vhid' bsln' db


vehIDLnData =
[(100,"37N"),(101,"37N"),(102,"37N"),(103,"37N"),(104,"37N"),(105,"37N"),(106,"37N"),
 (107,"37N"),(108,"37N"),(109,"37N"),(110,"37N"),(111,"37N"),(112,"37N"),(113,"37N"),
  (114,"37N"),(115,"37N"),(116,"37N"),(117,"37N"),(118,"37N"),(119,"37N"),(120,"37N"),
   (121,"37N"),(122,"37N"),(123,"37N"),(124,"37N"),(125,"37N"),(200,"38N"),(201,"38N"),
    (202,"38N"),(203,"38N"),(204,"38N"),(205,"38N"),(206,"38N"),(207,"38N"),
     (208,"38N"),(209,"38N"),(210,"38N"),(211,"38N"),(212,"38N"),(213,"38N"),(214,"38N"),
      (215,"38N"),(216,"38N"),(217,"38N"),(218,"38N"),(219,"38N"),(220,"38N"),(221,"38N"),
       (222,"38N"),(223,"38N"),(224,"38N"),(225,"38N"),(300,"37S"),(301,"37S"),
        (302,"37S"),(303,"37S"),(304,"37S"),(305,"37S"),(306,"37S"),(307,"37S"),
         (308,"37S"),(309,"37S"),(310,"37S"),(311,"37S"),(312,"37S"),(313,"37S"),
          (314,"37S"),(315,"37S"),(316,"37S"),(317,"37S"),(318,"37S"),(319,"37S"),
           (320,"37S"),(321,"37S"),(322,"37S"),(323,"37S"),(324,"37S"),(325,"37S"),
            (400,"38S"),(401,"38S"),(402,"38S"),(403,"38S"),(404,"38S"),(405,"38S"),
             (406,"38S"),(407,"38S"),(408,"38S"),(409,"38S"),(410,"38S"),(411,"38S"),
              (412,"38S"),(413,"38S"),(414,"38S"),(415,"38S"),(416,"38S"),
               (417,"38S"),(418,"38S"),(419,"38S"),(420,"38S"),(421,"38S"),(422,"38S"),
                (423,"38S"),(424,"38S"),(425,"38S")] :: [(Int, String)]
```