Wayfinding in scene space: modelling transfers in public transport

Rüetschi, U J

Abstract: One of the characteristics of public transport is the need for transfers. During transfers, travellers are pedestrian wayfinders and have to find their way from the place of arrival to the place of departure of another means of transport. Wayfinding in public transport takes place in two types of spaces: network space and scene space. Network space includes the transport network. Scene space consists of the halls, squares, and platforms at interchange nodes; it lacks an obvious network structure. Published timetables allow pre-trip wayfinding in network space, but wayfinding in scene space requires interaction with the actual environment, using the information provided therein. The theory of image schemata (cognitive patterns that structure our perceptions and actions) suggests that structural information is immediately usable. Other types of information, notably signage, require conscious effort to use. This research is about transfers in railway stations, i.e., wayfinding in scene space. I hypothesise that structural information provided by the architectural layout is enough to guide travellers through railway stations. Computational modelling is used to test this hypothesis. Image schemata form the cognitive basis for a formal model of scene space, called Schematic Geometry. A software agent is developed next and used to simulate wayfinding tasks in Schematic Geometry models of two railway stations. Results indicate that structural information supports wayfinding, but is not always enough to find optimal routes. Previews of parts of the environment are found to be highly effective wayfinding aids that build on human object recognition rather than on sign reading. Conclusions are that humans integrate information from various sources and of different types to achieve good wayfinding performance; well designed architecture can further support wayfinding by providing structural clues and previews. This thesis contributes to our understanding of wayfinding, especially the information needs for wayfinding in scene space, and provides a cognitively motivated but formal model of scene space, Schematic Geometry.

Wayfinding in Scene Space
Modelling Transfers in Public Transport

Dissertation
zur
Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät
der
Universität Zürich

von

Urs-Jakob Rüetschi
von
Suhr AG

Promotionskomitee

Prof. Dr. Sara I. Fabrikant (Vorsitz)
Prof. Dr. Klaus Dittrich
Prof. Dr. Werner Kuhn
Dr. Sabine Timpf (Leitung der Dissertation)

Zürich, 2007

Abstract

One of the characteristics of public transport is the need for transfers. During transfers, travellers are pedestrian wayfinders and have to find their way from the place of arrival to the place of departure of another means of transport.

Wayfinding in public transport takes place in two types of spaces: network space and scene space. *Network space* includes the transport network. *Scene space* consists of the halls, squares, and platforms at interchange nodes; it lacks an obvious network structure. Published timetables allow *pre-trip* wayfinding in network space, but wayfinding in scene space requires interaction with the actual environment, using the information provided therein. The theory of *image schemata* (cognitive patterns that structure our perceptions and actions) suggests that structural information is immediately usable. Other types of information, notably signage, require conscious effort to use.

This research is about transfers in railway stations, i.e., wayfinding in scene space. I hypothesise that *structural information* provided by the architectural layout is enough to guide travellers through railway stations. Computational modelling is used to test this hypothesis. Image schemata form the cognitive basis for a formal model of scene space, called *Schematic Geometry*. A software agent is developed next and used to simulate wayfinding tasks in Schematic Geometry models of two railway stations.

Results indicate that structural information supports wayfinding, but is not always enough to find optimal routes. *Previews* of parts of the environment are found to be highly effective wayfinding aids that build on human object recognition rather than on sign reading. Conclusions are that humans integrate information from various sources and of different types to achieve good wayfinding performance; well designed architecture can further support wayfinding by providing structural clues and previews.

This thesis contributes to our understanding of wayfinding, especially the information needs for wayfinding in scene space, and provides a cognitively motivated but formal model of scene space, Schematic Geometry.

Zusammenfassung

Charakteristisch für den öffentlichen Verkehr ist die Notwendigkeit des Umsteigens. Dabei werden alle Reisenden zu Fussgängern mit der Aufgabe, einen Weg zum Abfahrtsort des nächsten Verkehrsmittels zu finden.

Wegfindung im öffentlichen Verkehr findet in zwei Raumtypen statt: *Network Space* umfasst das Liniennetz; *Scene Space* umfasst die Hallen, Plätze und Bahnsteige der Umsteigeknoten und hat keine offensichtliche Netzwerkstruktur. Ein publizierter Fahrplan erlaubt *pre-trip* effiziente Wegfindung im Network Space. Wegfindung im Scene Space ist nur durch die unmittelbare Interaktion mit der Umgebung und der darin vermittelten Information möglich. Die Theorie der *Image Schemata* legt nahe, dass strukturelle Information unmittelbar verständlich ist; andere Informationstypen, allen voran die Beschilderung, bedürfen einer bewussten Anstrengung.

Mein Interesse gilt dem Umsteigen in Bahnhöfen und somit der Wegfindung in Scene Space, sowie der Hypothese, dass *strukturelle Informationen* des architektonischen Layouts ausreichen, um einen Reisenden durch einen Bahnhof zu leiten. Mittels *Computational Modelling* wird diese Hypothese getestet. *Image Schemata* dienen als Basis für ein formales Scene-Space-Modell, genannt *Schematic Geometry*. Ein eigener Software-Agent betreibt Wegfindung in Schematic Geometry Modellen zweier Bahnhöfe.

Die Ergebnisse dieser Simulationen zeigen, dass strukturelle Information zwar notwendig ist, im allgemeinen aber nicht ausreicht um optimale Wege zu finden. Die Gründe dafür werden analysiert und *Previews* als eine "halbstrukturelle" und äusserst effiziente Wegfindungshilfe identifiziert. Schliesslich kann festgehalten werden, dass Menschen verschiedene Informationen integrieren um erfolgreich zu navigieren, dass aber ein sorgfältiges architektonisches Design menschliche Wegfindung massgeblich unterstützt.

Diese Arbeit erweitert unser Verständnis von Wegfindung, speziell die dafür notwendigen Informationen, und sie stellt ein formales und kognitiv plausibles Modell für Scene Space, die Schematic Geometry, zur Verfügung.

Acknowledgements

In doing this research, I enjoyed support and encouragement, comments and critique, discussions and debates from and with many people who deserve my sincere gratitude.

Sabine Timpf advised my thesis and initiated and lead the project within which it was done. On the committee were Kurt Brassel, Klaus Dittrich, and Werner Kuhn. When Kurt Brassel retired, Sara Fabrikant took over as the responsible faculty member and brought this thesis to a successful end. Benjamin Kuipers reviewed my thesis and provided a copy of the "unpublished" Room Theory (page 62). The Swiss National Science Foundation funded most of the project.

My office mates David Caduff, Corinna Heye, and Sandro Bischof shared joys and endured woes. The whole GIS (Robert Weibel) and GIA/GIVA (Kurt Brassel/Sara Fabrikant) groups provided the necessary scientific background. Britta Allgöwer initiated valuable connections.

Participation at conferences and summer schools caused further encounters with many interesting people from around the globe. Georg Vrachliotis shared my interest for wayfinding and architectural design. With Alexander Wolff and Frank Schulz I had the chance to balance my bias towards scene space in the present thesis.

Christian Freksa invited me to visit the cognitive systems group in Bremen, which proved to be a most inspiring stay. Thanks are due to him and the whole cognitive systems group. Special mention deserves Kai-Florian Richter, with whom I had many discussions and co-organised a workshop on modelling environments. The interest by Johannes Schaub from the Swiss Federal Railways linked my thoughts and work to the "real world."

Finally, thanks are due to my parents for both material and immaterial support, and to Andrea for her genuine interest and lasting patience.

Table of Contents

Chapter 1

Introduction

Did you ever get lost in a railway station or in a public park? Probably not, because you are good at "wayfinding." But why are you good at wayfinding? What helps you getting around successfully? This research posits that human wayfinding heavily draws on structural information from the environment. Computational modelling will be used to examine this hypothesis for the specific case of wayfinding in railway stations.

1.1 Overview

Human activities take place in space. They are, usually, spatially separated from each other. Going from one activity to another is a conscious endeavour—a proper activity in its own right. This activity is called transport and requires wayfinding (the cognitive act of getting from here to there) and locomotion (the physical act of getting from here to there). The observable phenomenon is known as navigation on the individual level and as transportation on the aggregate level.

A specific feature of public transport is the frequent need to transfer from one means of transport to another. During transfers, all travellers are pedestrian navigators. My particular interest is with transfers in railway stations. Railway stations are a facility of the public transport system and constitute a designed environment for wayfinding. This environment is an instance of what shall be called *scene space* or a space lacking a network structure. It can be designed to facilitate or to hinder wayfinding.

Because work with actual railway stations (and other scene spaces) is not feasible, models are needed for experimentation. The goal of this thesis is to design such a model, argue for its validity, and use it to simulate wayfinding in scene space. The model shall describe the spatial semantics of the environment, that is, what its layout and furnishing mean to the traveller.

This requires the model to build on a solid cognitive basis as well as being formalisable, so that it can be implemented as a computer program.

The present thesis **contributes** to our general understanding of wayfinding in public transport systems, with a special focus on the interaction between the traveller and the structure of the environment. It will be shown that structure is not only a valuable source of information for wayfinding, but also that structure alone is not sufficient to guide a traveller efficiently to the intended destination. In practice, a balance is needed between good architectural design and good signage.

The intended **audience** is broad and varied. Those interested in *cognitive science*, especially in spatial cognition, wayfinding, and artificial intelligence, get new insights into the interplay between the environment and the reasoning processes of a mobile agent, along with a new way to represent spatial knowledge. Those interested in *architecture* will find a tool to represent and analyse designed space and its semantics for humans. Those interested in *transportation* will get a novel component for the simulation of transfers in multi-modal transportation. Finally, *geographers* as scientists of space will learn about a new perspective on how to structure space, a model that is different from the classical field, object, and network models.

1.2 Motivation

The motivation for writing this thesis rests on four pillars:

1. Public transport implies transfers, which expose the traveller to a complicated environment in which he/she has to find a route to some (intermediate) destination. Usually, this happens under tight time constraints, so wayfinding had better be good.
2. While wayfinding is a cognitive activity, it always takes place in a physical environment. How to sensibly model this environment is underresearched. However, good models are valuable, as they allow us to experiment with the environment and its effect on wayfinding.
3. Wayfinding is best understood as routing in a network—but what if there is no network? Simply imposing a network arbitrarily could easily create structure that contradicts human perception. What is really wanted is a formal reconstruction of spatial structure in accordance with principles of human cognition.
4. Image schemata are often used as a tool to analyse space and spatial meaning. Can they also be used for *synthesis*, that is, as primitive components from which to reconstruct an environment?

The focus on railway stations is motivated by their functional importance within a city [77:74] and because they are symptomatic for an environment where people navigate without personal navigation assistants and without an obvious network structure. Results are expected to generalise to other "scene spaces," such as shopping malls, public parks, and eventually larger urban areas.

1.3 Goals, hypotheses, approach

The overall goal of the present thesis is to analyse and model transfers in public transport, that is, wayfinding at interchange nodes in the public transport system. There are three subgoals:

1. Characterise public transport *as an environment* for wayfinding.
2. Build a formal but cognitively motivated *model of scene space* and test its fitness for use in wayfinding simulation.
3. Apply the model to analyse which environmental structures make human wayfinding in scene space so reliably.

The focus is on the environment and its spatial *structure*. Landmarks (salient elements of an environment [77,116]), although important for wayfinding in general, are only considered as far as they relate to the proposed structural investigation. Chapter 3 establishes this focus, substantiates the first subgoal, and serves as a premise on which the thesis builds:

Premise. *There are two radically different types of environments for wayfinding in public transport: those that exhibit a clear network structure and those without an obvious network structure.*

Environments with a clear network structure are called *network space*; those without an obvious network structure are called *scene space* because they are typically composed of individual scenes (Chapter 3). My interest is with scene space, because it is a new way of looking at space, whereas network space is the type of space that is traditionally investigated by the transportation community. This does not mean that network space is any less interesting or less complex than scene space.

The second subgoal, a model of scene space, builds on image schemata [54] and is therefore called *Schematic Geometry* (Chapter 4). Image schemata are not formalised in this work, but they provide the cognitive "grounding" (in the sense of Harnad's symbol grounding [43]) for Schematic Geometry. Describing a complete environment (or a non-trivial part of it) requires simultaneous consideration of many image schemata and their instances. Using

image schemata to describe complete spatial configurations was, to the best of my knowledge, never done before.

Hypothesis 1. *Image schemata can capture the semantics of complete spatial configurations (like the structure of a railway station), not merely of individual situations (like "the pawn is on the chess board").*

Image schemata structure human perceptual [54]. Since Schematic Geometry builds on image schemata, it also represents structural information in the environment. But it is not clear if this structural information is enough to successfully guide travellers through scene space.

Hypothesis 2a. *Structural information as encoded in Schematic Geometry is sufficient for successful wayfinding in scene space.*

Hypothesis 2b. *Adding signage and previews to Schematic Geometry accounts for the difference between optimal wayfinding and wayfinding using only structural information.*

Simulations in Chapter 5 provide the data that will be used in Chapter 6 to evaluate these hypotheses.

Even though "wayfinding" is frequently associated with "signage," some authors note that wayfinding is more than just signage [4,89]. In particular, architects (creating spatial structure through layout design) and signmakers have to work together [89]. While this seems obvious, it is worth investigating the relative importance of these two components on wayfinding performance.

My thesis is most closely related to Martin Raubal's thesis [100]. He had a clear focus on signage, assumed network space, used a graph to represent it, and designed a software agent to look for inconsistencies in the signage of an airport. My work aims to supplement Raubal's thesis by focusing on scene space and structural information. It remains to be discussed whether there is a natural link between network space and signage on the one hand and scene space and structural clues on the other hand.

The method used to investigate my hypotheses is computational (cognitive) modelling. This choice is motivated by the obvious infeasibility to "play" with the structure of real stations. Moreover, computational modelling offers these benefits over other methods: The model, once implemented, serves as (*i*) proof of concept, (*ii*) a platform for further experimentation, and (*iii*) can be used as a component in transport simulations. There is no human subjects testing. Instead, my approach builds on well-known properties of image schemata.

1.4 Thesis outline

Chapter 2 sets the context for this thesis, both from the perspective of transportation and wayfinding. It then introduces the concept of image schemata, the cognitive foundation for this thesis, and presents the method of computational modelling.

Chapter 3 proposes network space and scene space as the two main structural aspects of the public transport system with respect to wayfinding and discusses their properties. This chapter was presented at *Spatial Cognition IV* conference [107]. The remainder of this thesis is mostly concerned with scene space, not with network space.

Chapter 4 develops an abstract model for scene space, called Schematic Geometry. This involves defining cognitive schemata, explaining how they are instantiated, linked, and nested, and capturing this structure in a formal model. From this formal model, further concepts are defined, most notably the scene and the scene graph. Early ideas about Schematic Geometry can be found in [106]. Details about how Schematic Geometry captures spatial semantics are in [108].

Chapter 5 develops and tests a software agent that will find a way in a Schematic Geometry. While this is an interesting endeavour in itself, it mainly serves to test the hypothesis that most information for wayfinding in scene space comes from the schematic structure of the environment, and to test Schematic Geometry as a formal representation of scene space. A first sketch of the agent was presented at the *International Symposium on Transport Simulation* 2006 [109].

Chapter 6 discusses the results from Chapter 5, relating them to wayfinding in scene space, and evaluating the adequacy of Schematic Geometry as a model for scene space. It also considers alternative wayfinding strategies and information sources other than spatial structure.

Chapter 7 draws conclusions with respect to the stated hypotheses, to wayfinding, and to public transport in general. It identifies further research and points to other applications of Schematic Geometry.

Appendices contain additional material. Appendix A introduces partial ordering; its terms and concepts are assumed to be known and will be used in the main text without reference to this appendix. Appendix B explains how my implementation of Schematic Geometry encodes partial orders using a realiser. Appendix C presents the software tool that was developed for this thesis. Appendix D is a glossary of essential terms.

Chapter 2

Background

This chapter explains phenomena and theories that motivate and support this thesis: public transport, human wayfinding, image schemata, and computational modelling.

2.1 Understanding public transport

Public transport constitutes a *designed environment* for wayfinding, and consequently can be looked at from two perspectives:

the designer's perspective $\longrightarrow$ how is it conceived?
the traveller's perspective $\longrightarrow$ how is it perceived?

The designer's perspective is the view of the architects and engineers involved in the creation of public transport interchanges and stops[1] as well as the design of the network and the timetable. From the traveller's perspective, this environment is unalterable. It is up to the traveller to negotiate his way: he takes the role of a wayfinder who is given a task (reach some destination) in a particular environment (the public transport system). The ontological difference between these two perspectives was studied in [121].

Transportation, in general, refers to the movement of people, goods, information, and energy. The present thesis is interested only in the transportation of people using public transport. Amazingly, there is no widely accepted definition of public transport. But public transport is often characterised by certain properties [9]: public transport is the production of the service "transport" for masses of people, not just individuals; this service is completely fixed in space and time by means of the timetable; there is always a chauffeur, thus eliminating the need to drive oneself; and trips involve more

[1] The term "interchange" is used for large nodes in the public transport system, where several lines come together and possibilities for transfers exist, whereas "stop" refers to small nodes, typically serviced by only one line.

than one means of transport (including walking), that is, passengers have to change means of transport at stops (for entering and leaving the system) and at interchanges (also for transfers). Based on these properties, the following definition is proposed:

Definition. *Public transport is the set of* services *for the transportation of* people *according to a predefined* schedule *(fixing place and time) and subject to* published conditions of use*, employing* multiple modes *of transport.*

This definition is useful to tell public transport apart from other forms of transportation, but it does not consider how public transport is operated. Today, public transport is mostly organised according to the *line operation* principle, and this form of organisation is expected to remain prevalent in the future [78]. Brändli [10] defines *line operation* to be the servicing of a fixed sequence of stops with predefined departure times. This service is publicly accessible (subject to transportation regulations) and is bound to

1. the network of roads, contact wires, or tracks;
2. the lines, which use the road, track, or contact wire network;
3. the stops and interchanges; and
4. the timetable.

These bindings[2] influence how the traveller interacts with the system. They also indicate that public transport can be conceived of as a network only at a very abstract level. Every detailed investigation should respect that there are *lines* being operated *on* a network. It is true that the union of all these lines forms a network, but it is the timetable that decides if it is also perceived as well-connected network by the traveller.

Figure 2.1 shows how a journey using public transport is decomposed into spatially separated elements. The access, transfer, and egress (exit) nodes (together with interconnections, lines, tariffs, a schedule, and services) are all part of what is perceived as the public transport *system*. Because of the binding to stops and interchanges, travellers have to access (and leave) the system using some other mode of transport, like walking or cycling. Because of the binding to lines, transfers are inevitable for most trips. And because of the binding to a timetable, passengers are unlikely to access the system at arbitrary times;[3] rather, they have to plan in advance, look for services

[2] Original German terms: Strecke, Linie, Haltestelle, and Fahrplan.

[3] Brändli [9,10] notes that the arrival of passengers at stops is independent of the schedule if the stop is being serviced in intervals of no more than 7 minutes.

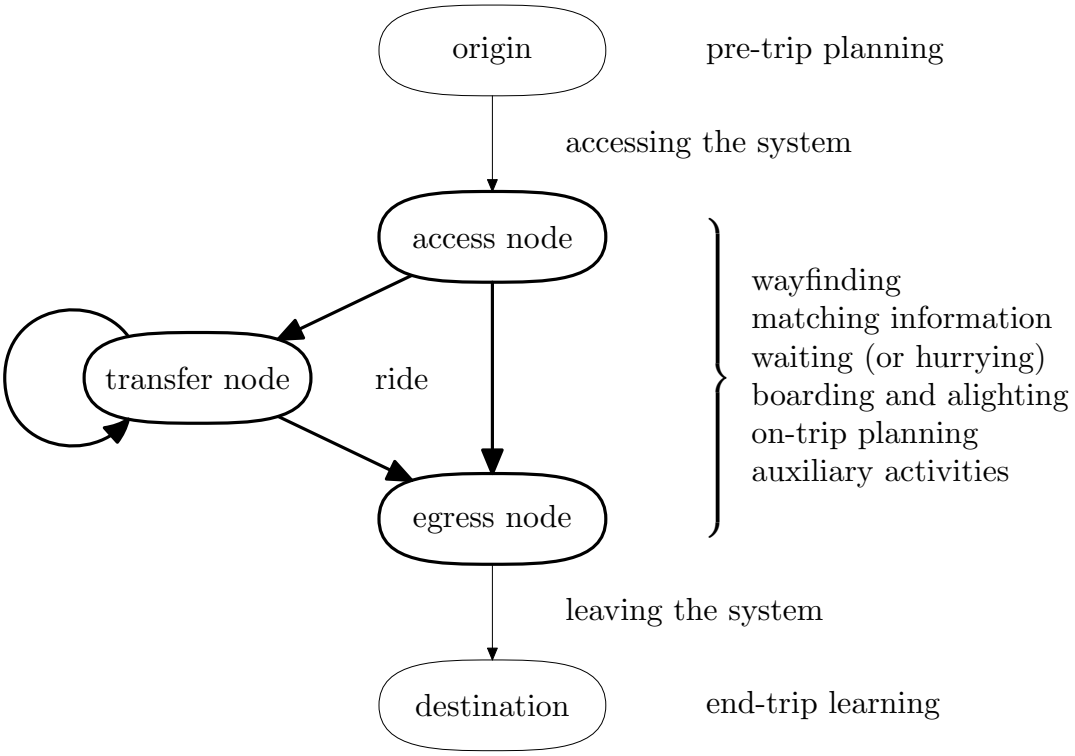


Figure 2.1 Elements of a journey using public transport (after a figure in [10]) with relevant actions and reasoning processes added. The separation into five different elements is typical of the line operation principle and not present in private car transport, which can take the traveller directly from origin to destination.

and connecting services, and try to optimise travel time, travel cost, route complexity [45], and other criteria [40].

The logistical framework of any journey consists of three phases or contexts [52] as follows:

pre-trip (also *pre-travel*): the planning context, in which people have to plan the trip while still being at the origin

on-trip (also *en-route*): the tracking context, in which people verify that the journey runs according to plan and in which adjustments may be made; this happens during the trip and especially at the access, transfer, and egress nodes

end-trip: after leaving the system, travellers assess the trip and have gained additional knowledge for future trips

These three contexts constitute the logistical framework for any journey. Nowadays, public transport operators provide useful tools in the Internet for pre-trip planning: interfaces to query the timetable, where the results usually are linked with other information such as services on board, accessibility to railway stations, weather conditions at the destination, and so on.[4] The on-trip context is the most time-critical, because the schedule imposes time constraints beyond the simple requirement to be at a destination at a certain time. No matter how well the pre-trip planning was done, there can be surprises on-trip, such as delays or cancellations of services. Moreover, even though information about the place of arrival and departure is usually available pre-trip, the station layout is usually not available other than by direct experience.

In summary, the public transport system constitutes a challenging environment for wayfinding, worthy for scientific investigation. It necessitates transfers between different modes and means of transport but makes it impossible to plan everything in advance.

2.2 Human wayfinding and models of human wayfinding

In 1960, Kevin Lynch published "The Image of the City," a study about how people understand the structure of a city. Lynch termed this the "imageability" of a city and found that five elements were common to all those city images: landmarks, paths, edges, nodes, and districts. This finding is important by itself and often-cited, but the book's most lasting impact is that it initiated a new research direction: wayfinding.

Wayfinding is navigation with a focus on cognition [21], whereas the physical component of navigation is often referred to as locomotion. We all accomplish many wayfinding tasks every day, such as going to work, shopping, or visiting friends. In the most general of terms, wayfinding is purposeful[5] [40:6] interaction [3,21] with an environment, the purpose being to reach a destination. Wayfinding is successful if the destination can be reached within spatial and temporal constraints and facing the unavoidable uncertainty involved with the task [3,4].

---

[4] A typical example can be found at `http://www.sbb.ch/`, the web site of the Swiss federal railways. It provides access to the timetables of virtually all transportation providers in Switzerland. This site was once reported to be the most-used Internet site in Switzerland, hinting at the huge demand for pre-trip planning in public transport.

[5] See [119] for an account on *playful* movement.

This section presents a selective review of the vast wayfinding literature, focusing on the *process* of wayfinding, the *cognitive map*, the *robustness* of wayfinding, and *computational models* of wayfinding.

Wayfinding as a situated process. Wayfinding can be understood as a problem-solving process, the problem being to get to some destination. This process is spatial, continuous (on-going while travelling), has to cope with uncertainty [4] and is always situated in an environment.

An important tool in this problem-solving process is the formation of a travel plan [34], whose execution results in actual behaviour. Travel plans are hierarchically structured sets of wayfinding decisions and therefore a mental solution to the problem [4]. Because of the uncertainty involved in wayfinding, the plan probably has to be revised while travelling, making planning a continuous process.

The hierarchical structure of travel plans allows for coarse initial planning and refinement only when needed. Wiener and Mallot call this the *coarse-to-fine planning* hypothesis and point out that it requires keeping a complete (but coarse) plan in mind [134]. Because this can be cognitively demanding, Wiener and Mallot alternatively posit a *fine-to-coarse planning* hypothesis. According to this hypothesis, wayfinders rely on the hierarchical structure of the environment and plan on a detailed level right from the outset, but only for the immediately following steps (compare with Section 5.2).

Allen [3] noted that there are three different wayfinding tasks: travel to a familiar destination, exploratory travel, and travel to a novel destination. Going to work is travel to a familiar destination and such a common task that it is performed largely without conscious thinking [38]; it is solved using *habitual locomotion*. Children exploring their neighbourhood qualifies as exploratory travel. Holiday travel is often travel to a novel destination. Allen lists wayfinding means that can be used to solve wayfinding tasks. Habitual locomotion is one of those means, but it only helps with travelling to a familiar destination. Other means include piloting between landmarks (requires knowledge about a sequence of landmarks) and path integration[6] (monitoring self-movement and using it to update the current location; ants and bees are very good at this). In specially prepared environments, following a marked trail is another wayfinding means (for example, colour-coded trails in large buildings), but it can easily become cognitively demanding (for

[6] Path integration is also known as dead reckoning, especially in navigation [51], where "dead" is a corrupted abbreviation for "deduced" [25:152,94].

example, at a complicated interchange with many signs). The last wayfinding means mentioned is reference to a cognitive map.

Downs and Stea classify travelling to a specific destination, novel or familiar, into four subtasks [25]: orientation (present location), route choice (plan a route to the destination), keeping on track (ability to follow the route), and discovering the destination (by recognising it). The important task of destination choice is assumed to already have been made. This sequence of subtasks nicely applies to travelling within the public transport system (Figure 2.1 on page 14). It should be noted that recognising the destination usually is a simple task, but not always: just imagine an overcrowded bus at night, with incomprehensible announcements of the stops!

While the tasks of destination choice and route choice can often be done pre-travel, keeping on track is an on-trip task that requires the integration of information in the environment with previous knowledge about the route and the destination. Weisman [133] and later Gärling et al. [34]found that environments are easier for wayfinding if they have a high degree of differentiation, provide good visual access, and have a low complexity of the spatial layout. These factors are also referred to as the *physical-setting variables* of wayfinding. Both studies also mention signage, Weisman as a fourth variable in addition to the previous three [133], the Gärling et al. as a means to improve wayfinding, especially for newcomers [34].

Wayfinding is always an interplay between internal cognitive processes and an environment. The more information the environment provides, the less knowledge is required on the wayfinder's part, thus increasing the usability of the environment. Donald Norman coined the terms *knowledge in the world* and *knowledge in the head* to describe these two sources of information [93]. However, simply filling the environment with information is not the solution to good wayfinding design, as it can lead to information overload [4:34].

More than a decade ago, Gluck suggested that wayfinding research should focus on the *information needs* instead of the products of wayfinding (behaviour and the cognitive map) [37]. When talking informally with people about information needs for wayfinding, they almost invariably mention signage. This thesis, however, examines the hypothesis that signage is just one source of information, along with the other physical-setting variables, especially the architectural layout.

Wayfinding is frequently assumed to take place on networks and this has implications for models about wayfinding, where graphs (as a formal representation of networks) are prevalent [14,74,100,123,124,135]. This can

be justified by the great importance of routes to human wayfinding and human life in general; indeed, humans exhibit what Kuipers referred to as "sequential behaviour" [66]. But there are also environments that lack an obvious network structure (see Chapter 3). In such environments, there is no clear relationship between routes and the environment.

Knowledge in the head is certainly very important and interesting, as the huge volume of research on the cognitive map illustrates. But a wayfinder's environment is simply too complex to be completely known [15], let alone uncertainties such as cancelled services in public transport. In this sense, wayfinding is a problem-solving process that is situated in an uncertain environment. Despite the environmental focus in the present thesis, basic knowledge about the cognitive map and cognitive mapping is indispensable as it hints at which environmental features are essential for wayfinding.

The cognitive map. The "cognitive map" records spatial information about the world around us. It is knowledge about large-scale space [65]. The term was first used by behavioural psychologist Edward Tolman, who was working with rats [125]. He found that rats that were trained to take a particular way to get food would find an efficient alternative if the trained way was suddenly blocked and concluded that this remarkable behaviour can only be explained if rats have a map-like representation of space, not simply with learned stimulus/response patterns [125]. The idea of map-like representations in the mind is at least as old as a 1913 paper by the psychologist Trowbridge [127].

It was, however, soon realised that the cognitive map cannot be a cartographic map: a complete and consistent representation of spatial information in one reference system. Rather, the cognitive "map" is an incoherent and incomplete collection of knowledge fragments [130], for otherwise observed behaviour could not be explained. For example, humans were found to be able to pilot from landmark to landmark to get from A to B, but not in the other direction [77], so knowledge about routes seems to be stored asymmetrically. One of the best-known experiments is by Stevens and Coupe [118]. They found that most people think that Reno (in the state of Nevada) is east of San Diego (in the state of California), which is not true, even though Nevada as a whole is east of California. Therefore, they conclude, there seems to be a hierarchical encoding of spatial information that leads to this misjudgement. Further evidence for hierarchies in spatial knowledge is given in [49] and [83]. Tversky [129,130] describes two other phenomena, called the alignment effect and the rotation effect: she found that spatial features were often rotated and shifted so that they are nicely aligned. For example,

the United Stated and Europe are often aligned (in fact, Europe is north of the United States) and South America is rotated and shifted so that its west coast is aligned with the west cost of North America (in fact, South America's west cost is significantly east of North America's west coast). Portugali's edge effect [98] accounts for the strong organisational effect of edges like a coastline.

These findings resulted in a plethora of alternative terms for "cognitive map," such as "cognitive collage" [131] or "cognitive atlas" [48,67]; see [60] for an impressive list. Nevertheless, "cognitive map" still is the most frequently used term and if we all accept that it is only a *metaphor* for fragmented and distorted spatial knowledge, then this is fine and even useful to avoid terminological confusion. This work continues to use the term.

Lynch introduced the method of sketch mapping to elicit what is in a person's cognitive map of an environment [77]. He found that five elements are used over and over again:

Paths. Channels of actual or potential movement. Predominant in the cognitive maps of many people.

Edges. Linear elements not used/considered as paths; breaks in continuity; examples: park boundary, coastline, walls. Can serve as boundaries for districts. For many people, they are an important organising feature.

Districts. Sections of a city with a common and identifying character. They are always identifiable from within, but not necessarily from the outside. Together with paths the dominant element.

Nodes. Strategic spots with a radiating influence, typically junctions or more generally concentrations of any kind, can be entered by the observer.

Landmarks. Like nodes point-like features, but always external to the traveller and not enterable (though often approachable). Increasingly relied upon as a trip becomes more familiar.

Surprisingly, there are no "gateways" or "links," that is, elements that connect two spaces such as a bridge over a river. Probably, these were conceived of as paths. However, Chown convincingly argued for the importance of "gateways" in understanding ("parsing" in his words) an environment [16].

Lynch's study gives an insight into what constitutes a cognitive map, but it does not tell how this knowledge is learned and organised. According to Siegel and White, spatial knowledge develops in three stages [114]:

1. landmark knowledge (what there is in the world)
2. route knowledge (ordering of landmarks along routes)

3. survey knowledge (landmarks and routes arranged in two-dimensional reference frames, allowing for metrical judgments shortcut finding)

While these three types of knowledge are still accepted as types of spatial knowledge, the strict sequence was critisised based on empirical findings [53, 87]: humans, children as well as adults, that are exposed to a new environment, acquire knowledge of all three types simultaneously.

Robustness. Chown hints at the robustness of human wayfinding [15]. It is indeed amazing how effective we are at wayfinding, even with minimal knowledge, in a constantly changing world, when our attention is with something else, like having a conversation while changing trains. How is this possible? Unfortunately, there is hardly any research that investigates the robustness of the wayfinding process.

Chown argues that wayfinding is robust because it (*i*) uses qualitative representations in the cognitive map and (*ii*) relies on the environment [15]. He illustrates these claims with a direction-giving example: "go straight until you come to a river, then follow it to your left [until you can see the goal]." This direction is imprecise and qualitative. It relies on the environment and humans' excellent object recognition (we easily recognise the river and the goal) [15,17]. If the directions were instead "go 507.5 meters at a heading of 45.78," even a small amount of imprecision—either in the instruction or its execution—could result in missing the goal.

The PLAN model of wayfinding [17] posits that there are two systems involved in wayfinding, the "what system" (object recognition) and the "where system" (spatial relations). Excellence in one system can compensate for a deficit in the other: humans' excellent "what system" compensates for a comparatively poor "where system." But the "what system" requires a relatively constant world: objects that we cognised once need to remain in the world such that they can be re-cognised later on.

Qualitative representations and reasoning are the other important ingredient for robust wayfinding. Qualitative (spatial) reasoning is an entire research field. The term "qualitative" is used to mean symbolic, discrete, and behaviour-relevant [19]. It should be noted, however, that qualitative reasoning is not only an alternative to quantitative reasoning, it is often the only option, as Freksa's Aquarium Metaphor illustrates [30].

With a focus on learning, Kuipers presents an interesting thought experiment about the seemingly imperfectly engineered cognitive map with all its peculiarities and distortions [68]. He asks "could it have been any other way?" and concludes that "No. The constraints of learning a large-scale environment from local observations while operating under interruptions and

resource limitations are strong enough that any process that achieves the required level of performance must resemble the human cognitive map."

In conclusion, qualitative reasoning/representation does not only help with robustness, it is often the only way of reasoning and representation (Freksa's aquarium metaphor [30] and Kuipers' cognitive map thought experiment [68]). This leaves us with Chown's reliance on the environment and serves as another justification for my focus on the wayfinder's environment.

Computational models of human wayfinding. Several models exist that try to cast human wayfinding into a computer program. Kuipers' TOUR model [64,65] simulates spatial learning, that is, building a cognitive map. The cognitive map consists of routes (sequences of view/action pairs), which can be integrated into a topological network of places and paths, as well as regions, boundaries, containment relations, and eventually a metrical map. These representations build a hierarchy. Simple observation of view/action pairs (so-called sensorimotor schemas) are assimilated into route descriptions, from which not only a network is derived, but also boundary relations (left-of, on-path, right-of) and regions. The metric map is derived from turn and travel actions by augmenting them with a turn angle and a travel distance. This hierarchy was revised and stated more explicitly in SSH, the Spatial Semantic Hierarchy [69], a theory of spatial representation and learning. The main virtue of the TOUR model is to illustrate this learning of complex spatial knowledge from simple sensorimotor inputs. The environment that is being learned is completely abstracted behind the views and actions, giving the model a lot of generality.

Several models are specifically tailored to learning spatial networks: ELMER [82], TRAVELLER [74], and NAVIGATOR [41]. NAVIGATOR builds on empirical results and uses heuristics to find his way. As with TOUR, the environment is learned by navigating it. TRAVELLER focuses on route planning: based on the assumption that the relative locations of origin and destination are known, a search process starts from both origin and destination to find a route. ELMER is similar but mixes route planning and plan execution (as in the wayfinding process described by Arthur and Passini [4]).

None of the models reviewed so far takes into account the immediate use of cues in the environment for wayfinding. This is done by Raubal's computational model of an agent finding a way in an airport [100]. The environment is again a network, represented by a graph, but the agent makes immediate use of information provided by the environment. Information in the environment originates from signs that indicate the paths to gate areas and gates in the airport. The agent matches this information against its

internal representation of the goal (a specific gate) and uses this to decide along which of the outgoing edges of the current node to go. When several paths lead towards the destination (according to the signage at a particular decision point), then the agent chooses somewhat arbitrarily the one that goes into the most preferred direction. If no relevant information is available at a decision point, then the agent reports this and gives up (a real human would probably ask or choose randomly and start exploring).

2.3 Image Schemata

Image schemata originated in philosophy and linguistics. They refer to recurrent cognitive patterns that help us making sense of our perceptions and actions. This section reviews the image schemata literature, mostly from a geographic information science point of view, and introduces all those image schemata that build the foundation for Schematic Geometry (Chapter 4).

The origin of image schemata. Philosopher Mark Johnson developed the notion of image schemata in the late 1980ies [54], drawing on work in cognitive linguistics, most by Len Talmy, Ron Langacker, and George Lakoff (e.g., [72,73,120]). According to Johnson, image schemata are recurrent cognitive patterns that structure our perceptions and actions. They are abstracted from bodily experience and independent of concepts. Image schemata consist of parts and relations that can capture the structure of perceptual images and thus help in the translation from sensory input to meaningful knowledge about the world.

For example, the CONTAINER image schema embodies the idea of containment, separates an inside from an outside, stipulates that there is a boundary between the inside and the outside, and induces several (spatial) relations, like being inside and going into. Whenever the CONTAINER image schema is invoked, it gives meaning to language (like the preposition into) or visual perceptions, and helps with establishing analogy between different situations that also invoked the CONTAINER image schema: once a child realised that his toys can be put into a box, it is immediately obvious that his pencil can be put in his desk's drawer later on. Finally, containers can be inside containers and the induced relation is transitive [54:22].

Other examples include: ATTRACTION, the general idea of a force that pulls two entities together, both in a physical and in a social sense; PATH, a source that is connected with a goal through some trajectory, defining both location in space and a direction of movement; COLLECTION, a set (in the mathematical sense) of things belonging together (though the schema does

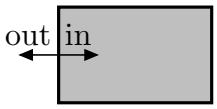
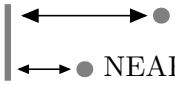
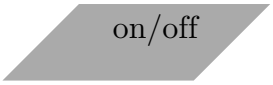
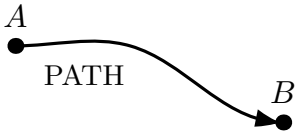
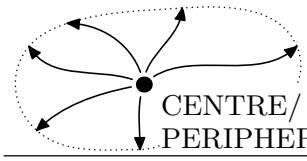
not state why they belong together); and many more. Table 2.1 lists all image schemata that are of importance to the present thesis.

Johnson provides a non-exclusive list of image schemata [p. 126] and describes some in detail; further characterisations of some image schemata can also be found in [72:271–275]. It is also fine to introduce new image schemata, as long as they qualify as such, that is, they are bundles of components and relation that add structure to the world around us. Many of the image schemata are inherently spatial and thus of great interest to the geographic information science community. In fact, two authors even see image schemata as "the fundamental experiential elements from which spatial meaning is constructed" [29].

The idea of schemata in human cognition was not entirely new when Johnson introduced image schemata. Ulric Neisser used schemata in 1976 as a component in the perceptual cycle [90]: schemata accept perceptual input and direct movement, and they are modifiable by experience [p. 54]. Neisser also emphasises the role of schemata in wayfinding and navigation [p. 111]. Johnson, however, stresses the imagistic and non-propositional nature of image schemata. He sees image schemata primarily as the abstract structure of images [54:*xix*] (whence the name). Also important to Johnson is that image schemata are embodied, that is, they originate from pre-conceptual bodily experience, from our interaction with the physical world around us.

Although Johnson is a philosopher, his image schemata became immediately popuplar in linguistics. This is probably best illustrated in the book "Women, Fire, and Dangerous Things" by his colleague George Lakoff [72]. Another notable work from linguistics is Dewell's 1994 study of the preposition "over" in terms of image schemata [23]. Dewell recasts the earlier studies of the same preposition by Brugman [11] and Lakoff [72] in a more rigorous way. He starts with a "central schema," an arc-shaped trajectory (instead of the flat "across" trajectory used by Brugman and Lakoff) over an object and manages to explain countless applications of the word "over" by subjecting this central schema to a number of image-schema transformations: multiple trajectories, multiplex-mass, segment profiling (center, downward, upward, freeze-frame at peak), resulting state (roughly what Lakoff called the "endpoint focus"), trajectory-part profiling (e.g., extending-path trajectory), and shifted perspective. Using these "variations on a theme," Dewell manages to give an overwhelmingly complete description of the meaning of "over."

Table 2.1 Important image schemata used in this thesis

Image Schema	Description and Locatives
 CONTAINER	The concept of containment, defining an inside and an outside, divided by a boundary; containers can be nested and the induced relation is transitive. open, closed, inside, in, out of, into outside
 NEAR/FAR	A qualitative idea of distance with two vague categories, near and far; always immaterial. next to, close, beside, near, between
 SURFACE	A surface in the colloquial sense: a plane to put things on, like a wall on which a picture can be hung or a floor on which a table can be put. on, off, under, above, below, level
VERICALITY 	An abstraction of the experience of gravity, domain of "up/down" relations, home of the "more is up" and "less is down" metaphors. up, high, top, down, low, bottom, rising, falling
	A (directed) trajectory that connects a source to a goal; very often used metaphorically as in a "long way" to achieve a goal or even the "path of life." front, back, left, right, sideways, end
	Mutual connection: two entities and a bond in between; the induced relation is symmetric and if either entity is destroyed, then so is the link. together, joined, separated
	The concept of centrality, a core which is the centre of a gradient, in geographical parlance a nodal region (as opposed to a uniform region) [7]. around, middle, corner, opposite
COLLECTION	A group of things that belong together for some reason that is not stated, mathematically a set.
OBJECT	A discrete entity in space (colloquial sense).

Locatives and the ordering of the table from most basic (top) to to most advanced (bottom) are from [32]; COLLECTION and OBJECT were not part of that study but are important to the present work.

Image schemata in geographical information science. Image schemata have some tradition in geographical information science, where they were used for such diverse purposes as the analysis of spatial language [32,80], the design of user interfaces [63], information visualisation [27], investigations of spatial relations [105], and wayfinding [102,103,104].

A 1989 paper by David Mark [80] is probably the first publication in the field of geographic information science that uses image schemata. Mark notes that most Indo-European languages express spatial relations through prepositions and gives the example "I was standing *in* my back yard *on* my property *in* Amherst." The choice of preposition, Mark claims, depends on the image schema adopted: back yard is a CONTAINER, demanding the preposition "in," whereas the property is a PLATFORM (Johnson would probably have called it a SURFACE), demanding the preposition "on."

An interesting study that also relates to language is that of Freundschuh and Sharma [32]. They analysed children's story books for locatives (like inside, open, off, behind, inside, across, near, left, etc.), related them with spatial image-schemata (see Table 2.1) and found that CONTAINER, NEAR/FAR, and SURFACE are among the most basic image schemata in that they are learned by children at a very early stage. The image schemata LINK and CENTRE/PERIPHERY turned out to be the most advanced (that is, learned at a later stage) of those schemata investigated.

Table 2.1 lists image schemata that have not yet been explained. SURFACE is a surface in the colloquial sense, that is, a plane to put things on, like a wall on which a picture can be hung or a floor on which a table can be put. VERTICALITY is an abstraction of the experience of gravity, the domain of "up/down" relations and home of the widespread metaphor "more is up" and "less is down." LINK is the idea of a mutual connection, consisting of two entities and a bond in between; the induced relation is symmetric and if either entity is removed, then so is the link. CENTRE/PERIPHERY is the concept of centrality, a core which is the centre of a gradient, in geographical parlance a field or a nodal region (as opposed to a uniform region) [7]. NEAR/FAR is about closeness and therefore closely related with CENTRE/PERIPHERY and LINK, but NEAR/FAR is always immaterial, a vague qualitative relation. OBJECT is an object in the colloquial sense, a discrete entity in space, and COLLECTION is a group of things that belong together for some reason that is not stated, mathematically speaking a set.

Image schema related work of a different kind is the Container-Surface algebra by Rodriguez and Egenhofer [105]. Starting from several case studies involving a box, a ball, a table, paper, a pencil, and a room, they derive an

axiomatisation of operations like "move into/onto" and "remove from" and analyse their transitive properties. The elements that participate in these operations are seen as OBJECT, SURFACE, and CONTAINER image schema instances and provide a cognitively solid basis for the proposed algebra.

Finally, there is Raubal's work on wayfinding in airports, which is collected in [101] and resulted in his Ph. D. thesis [100]. Of particular interest to this section about image schemata is an early study [103] where people were given the task of going from the departure hall to the gates, shown pictures along that way, and interviewed about the spatial experience that these pictures evoke. These interviews were then analysed and it was found that the image schemata from the Freundschuh and Sharma study [32] were used, along with many others, less spatial image schemata, including ATTRACTION (people were especially attracted by signs) and BLOCKAGE (a pillar is blocking the view). Superimpositions of image schemata [54:125] were frequently found, for example, when someone moves to the ticket counter, then there is a PATH implied that creates a LINK between the present position and the ticket counter, and there is also a SURFACE on which the movement takes place. The authors conclude that image schemata, obviously present in human experience, should also enter the design process.

All the work presented so far used image schemata as an analytical tool to add structure to an otherwise amorphous phenomenon: Dewell used them to analyse the meaning of "over," Freundschuh and Sharma used them to group locatives according to their spatial meaning, and Raubal used them to analyse interviews. Rodriguez and Egenhofer's Container-Surface takes a different approach: they use image schemata at the foundation of a new formalism, that is, in a constructive way. To the best of my knowledge, however, nobody used image schemata as building blocks for the *synthesis* of space, although [103] suggests this should be done. Chapter 4 follows this suggestion and builds a formal model of scene space, based on image schemata.

Formalisation issues. There are some works that try to formalise image schemata. Kuhn and Frank presented an algebraic approach [63], but it turns out that CONTAINER and SURFACE are isomorphic, that is, they have the same structure (Figure 2.2). This not only contradicts the structure-identifying property of image schemata, it also means that the two schemata cannot be differentiated within the algebraic formalisation. Raubal and Egenhofer use a notation where the image schemata appear as predicates

Sorts	Surface, Item, Boole
Ops	new: $\rightarrow$ Surface
	put: Surface $\times$ Item $\rightarrow$ Surface
	on: Surface $\times$ Item $\rightarrow$ Boole
Eqs	on(new, i) = false
	on(put(s, i_1), i_2) $\equiv$ ($i_1 = i_2$) ? true : false

Figure 2.2 An algebraic approach to formalising the SURFACE iamge schema [63]. The algebraic specification for CONTAINER turns out to be isomorphic (substitute "in" for "on"), that is, having the same structure. Since the algebraic approach of formalisation captures *only* the structure of an object, CONTAINER and SURFACE cannot be distinguished within the proposed formalisation.

over the objects involved [102]. They do not, however, state any properties of and relations among the schemata, leaving the approach purely notational. A later paper by Frank and Raubal attempts to formally specify image schemata by focusing on a formalisation of spatial prepositions, but ends up in stating that "the current approach trying to capture image schemata with the definition of spatial prepositions is too limited" [29].

It should not come as much of a surprise that image schemata defy formalisation, for their two key features are their image-like and non-propositional nature [54:*xx*]. Moreover, Johnson never gave a precise definition of image schemata. This does not mean that it is impossible to formalise image schemata, but it is certainly going to be a major endeavour. For the purpose of my thesis, I will not try to formalise image schemata, but rather use them as a bridge between human spatial concepts and formal representations (Chapter 4).

What about affordances? Perceptual psychologist James J. Gibson introduced the concept of affordance to cope with the immediate usability of things in the environment [36]. "The affordances of an environment are what it *offers* the animal, what it *provides* or *furnishes*, either for good or for ill" [p. 127]. This concept is very different from image schemata! While image schemata tell us something about the structure and the meaning of our perceptions and actions, affordances tell us about our options for immediate action without much reasoning. Image schemata are a cognitive concept, but affordances are on the perceptual level. The two concepts complement each other in a very useful way, as will be shown in Section 4.1.

The most comprehensive account of the theory of affordances from a wayfinding point of view can be found in Martin Raubal's Ph. D. thesis [100]. The first paper that uses image schemata and affordances together in an

```
Object -> car "little red car"
  with name "little" "red" "car "kar1",
       description "Large enough to sit inside.
                    Among the controls [...]
       [...]
  has switchable enterable static container open;
```

Figure 2.3 The two best known design systems for interactive fiction games (computer adventure games that are purely based on textual input and output) are Inform and TADS. Both systems employ image schemata and affordances, most likely without their authors knowing about the scientific nature of these concepts. The figure shows a fragment of Inform code with keywords that refer to image schemata and affordances underlined. These keywords are used by game authors to tell the game engine about essential properties of objects and the game engines derives potential player activities but also physical properties such as lighting.

investigation of wayfinding is by Raubal and Worboys [104]. A noteworthy discovery is that Inform [91] and TADS (`www.tads.org`), two design systems for interactive fiction (text-based computer games), use both concepts in a very practical way. An Inform example can be found in Figure 2.3.

2.4 Computational Modelling

Chapter 5 develops a software agent that will exercise the model for scene space developed in Chapter 4 by using it as an environment for wayfinding. It also serves as a test for the hypothesis put forward in Section 1.3 that most information for wayfinding in scene space can be derived from the spatial layout.

This software agent is an instance of a computational model, that is, a computer program to simulate a cognitive process. The validity of this approach is the basic assumption behind cognitive science, a new research paradigm that replaced behaviourism in the 1950ies [5]. The novelty of this "cognitive revolution" [85] is that human behaviour is no longer explained in terms of stimulus/response relationships, but in terms of an information processing machine: the human mind is no longer treated as a black box; it is looked into and the tool for doing so is the analogy to the emerging digital computer.

The proposed analogy between a modern computer and the human mind justifies computational modelling as a new approach for studying the human mind. The term "computational modelling" has widened in scope beyond the human mind: "Computational models are created to simulate a set

of processes observed in the natural world in order to gain an understanding of these processes and to predict the outcome of natural processes given a specific set of input parameters" [46] To stress the focus on the human mind, the term "cognitive modelling" can be used instead and there is a whole conference series devoted to the topic [22].

The analogy between the human mind and the computer proved to be useful but it has its limitations if it is taken too seriously: if the human mind is thought to be an algorithm. This extreme view is referred to as the cognitivistic paradigm [97], a pejorative term. The human mind is not an algorithm, mainly because an algorithm is a purely abstract concept, whereas the human mind is always embodied. Theories like that of image schemata (Section 2.3) even state that knowledge directly draws on the embodiment of the human mind. Another difficulty with the cognitivistic paradigm is that computers and algorithms are designed to find optimal solutions, whereas humans are often satisfied with a good (but not optimal) solution, a phenomenon that is called *satisficing* [115]. These findings resulted in embodied cognitive science (or new AI), a reaction to the cognitivistic paradigm [97].

This does not mean that computational models are not useful. Rather, if they reproduce the human mind with its limitations, instead of searching for optimal solutions, they give hints at the kinds of errors humans likely make. Pfeifer and Scheier [97] mention four key advantages of computational models as compared to verbal descriptions or charts: by virtue of being computational, they are (1) inherently precise and (2) clear in their assumptions, it is (3) easy to assess their internal validity, that is, how well they implement the theory they are supposed to implement, and (4) they ease the communication among scientists (at least, if the underlying formal language is known). Kuipers even claims that "computational models provide the most productive view of cognitive processing available in psychology today" [67:9].

Chapter 3

Network Space and Scene Space

Wayfinding always takes place in an environment. Chapter 2 revealed that the wayfinding literature focuses on the process of wayfinding (especially on building and using the cognitive map) more than on the environment in which wayfinding takes place. Not surprisingly, most exceptions to this general rule come from people involved in architecture, such as Lynch, his disciple Appleyard, as well as Arthur and Passini.

A useful device when dealing with the environments of wayfinding is a classification of environments. For example, driving a car through a street network is a completely different task from navigating a boat across an ocean. This chapter proposes a classification of environments based on their prevalent *structure* as network space or scene space. The development of this classification was motivated by a study of public transport. The classification is, however, general enough to be applied to all environments.

3.1 Classification of spaces

Several classifications of space have been proposed. Most of them are based directly or indirectly on the space's size relative to the human body. For example, my desktop is a relatively small space and easy to overlook and manipulate, whereas my neighbourhood is a relatively large space that cannot be overlooked at once and hardly manipulated. None of the existing classification schemes is based on the structure of the space.

Downs and Stea distinguish between small-scale (perceptual) space and large-scale (geographic) space [25:197–199]. Small-scale space contains manipulable objects (like a pencil on a desk—whence the name table-top space), whereas large-scale space is so large that it cannot be perceived from one perspective, let alone manipulated. The observer is always immersed in large-scale space, but never immersed in small-scale space.

Kuipers uses a similar classification, also using the terms small-scale space and large-scale space [65]. Large-scale space is space whose structure cannot be observed from a single viewpoint (p. 129), that is, the "spatial structure that is larger than the sensory horizon of the agent" [70:40]. Small-scale space is defined to be not large-scale space. This implies that the observer can also be immersed in small-scale space in Kuipers' sense, unlike the interpretation of Downs and Stea [25].

A different classification was proposed by David Zubin and reported in [80:13–17]: based on the perception of (abstract) spatial objects, he defined four categories named Type A, B, C, and D:

A: objects smaller than the human body, viewable from a single perspective, for example a pencil on a desk

B: objects larger than the human body that cannot be seen from one perspective (but inferred), such as an elephant, the outside of a house, or a mountain.

C: "scenes" that can be perceived from a single vantage point, but only by scanning; examples: a room, a small valley, the horizon.

D: "territories" such as forests, cities, countries, or the inside of a house that cannot be perceived as a unit, but where small portions can be perceived as type A or type C objects.

The interesting thing to note here is that the sequence from A to D does not imply an increase in the size of the object or space. An explicity statement about the size of objects is only made for type A and type B. Type C objects can easily be much larger than type D objects, for example, the outside of a house (type C) versus the inside of a room in that house (type D). In particular, type C objects can be extremely large (like the horizon), but they still belong into Kuipers' small-scale category.[7] Only type D spaces qualify as large-scale space in the sense of Kuipers. In this sense, Zubin's classification is a refinement of small-scale spaces, but corresponds with large-scale space.

Montello distinguished four psychological spaces [86]. Figural spaces correspond to Zubin type A spaces; Montello also notes that picturial representations of potentially much larger spaces qualify as figural spaces. Vista spaces correspond with Zubin type C spaces or scenes: they can be scanned

[7] This is a hint that outdoor spaces are particularly demanding for robotics: isovists [6] can be huge indeed. Indoor and enclosed urban environments constrain what can be perceived: isovists are rather small.

from one vantage point by looking around. Environmental spaces also surround the observer but cannot be perceived without exploring, that is, moving around. Finally, geographic spaces are defined to be much larger than environmental spaces, so large, that it is not only impossible to scan them from one vantage point, but that they even cannot be experienced directly. The lack of direct experience can be compensated using symbolic representations such as maps.

The last classification of space to be mentioned here is by Freundschuh and Egenhofer [31]. They used three binary criteria (manipulability, the necessity to locomote to apprehend the space, and the size of space). Based on combinations of these properties they defined six types of space and relate them to earlier classifications. The novelty with this classification is that it brings several criteria together and states them explicitly, whereas the classifications by Zubin and Montello directly define four classes but do not give strict rules for classification.

The article by Freundschuh and Egenhofer [31] reviews some more classifications, or, as they call them, cognitive models of space. However, the tenor of all those models is well captured with those just reviewed.

3.2 Networks and scenes

Many environments in which humans wayfind can be conveniently abstracted as networks (and more formally as graphs): structures consisting of nodes that are interconnected by edges. Examples include a city's street network, the lines operated by a public transport provider, and the paths in a park. I refer to such environments as **network space**.

Other environments consist of open spaces or "scenes," such as halls, areas, and squares. These scenes are hierarchically grouped and connected with one another, but there is no obvious network structure. I refer to such environments as **scene space**. Examples include public greens, university campuses, shopping malls, and train stations.

Definition. *A* scene *is what can be seen by looking around but without significant walking about and forms a coherent and obvious entity in large-scale space.*

A scene is not defined purely by visibility and therefore is different from an isovist [6]; equally important is reachability: a scene stops where a line of sight crosses a boundary such as a window or a barrier which travellers are not supposed to trespass. For example, a scene is:

a room, but not the part demarcated by a carpet;

a station square, but not the tramway stop on it;
a platform in a station, but not the adjoining platforms.

Wayfinding in public transport takes place on traffic networks. These consist of lines that are interconnected at nodes, ranging in size from small stops to large railway stations. The network is essential to wayfinding because it is the basis for routing decisions. Today, such routing is much simplified by the presence of electronical timetables in the Internet. But to the traveller, the stops and stations are equally important, for they are the places where the complexity of the system is most intensely experienced, and they are where there is usually no network that helps with routing. Therefore, wayfinding in the public transport system involves both, network spaces and scene spaces, and most evidently so at large interchanges such as big railway stations.

When considering interchanges in the public transport system, it is easy to see that network space is a crucial determinant of human behaviour in scene space. This is because network space aspects of an interchange node determine (by way of the timetable) where and when means of transport arrive and depart. The *constraints* thus imposed are characteristical of public transport.

An example. Suppose you have to travel from Bern (the Swiss capital) to Oerlikon (a Zürich suburb) and your pre-trip query of the timetable gave you this information:

Bern	dep	07:47	track 6	} IC 911
Zürich main station	arr	08:56	track 13	} IC 911
Zürich main station	dep	09:06	track 21/22	} S5 18530
Zürich Oerlikon	arr	09:12	track 6	} S5 18530

Your train arrives at 08:56 (on time) in Zürich main station. The second leg of your trip consists of a short ride using the "S5" rapid transit railway, departing at 09:06 from track 21/22. You walk towards the platform's darker end, because that is where you expect the station building to be. Moreover, this is the direction where most people are going.

On your way along the platform you diligently confirm the departure information (09:06 from track 21/22) using a nearby departure screen. It still holds true. But the track designation "21/22" is a bit disturbing, because you expected a simple number like "7" or "22," not what looks like a fraction. Your strategy of following the crowd towards the darker end of the platform in this cul-de-sac station is successful: you eventually find a sign with "21/22"

on it, pointing down an escalator. After some more turns and escalators, you finally reach the connecting train, somewhere underground.

This example shows that information is used from various sources. It can be classified into two main aspects of the public transport system:

1. Information about the network and the timetable:
   - Internet query about the route, departure and arrival times, and service names (like "S5"), given an origin, a destination, and a desired time of arrival
   - Local system information: the departure board contains a locally relevant subset of the overall timetable
2. Information about the layout of the interchange:
   - Local system information: the departure board also links timetable information to information about the interchange node: given a departure time and a place goal, the precise location of departure can be found
   - Signage: the symbols for guiding travellers and indicating where things are
   - Architectural layout: the combination of elements like platforms, halls, departure boards and screens, escalators, as well as their accidental properties (e.g., "the darker side")

The division into information about the network/timetable and information about the layout of the interchange is motivated by the public transport system and how it presents itself to the traveller. Information of the first type is mediated (through the Internet and more traditionally through a printed timetable). Information of the second type is not mediated: it is obtained on-trip and in direct interaction with the various "scenes" of which the interchange is composed.

Two aspects of space. Trying to find mnemonic names for the two types of spaces characterised by the scenario, one may settle on "network space" and "scene space." The two spaces are defined as follows:

Network Space has a node/link structure. It is created by network and schedule engineers, and reflects historical, social, and economical processes. Network space is a mediated space, presenting itself by means of maps and schedules, announcements, and sometimes also delays.

Scene Space consists of scenes in a hierarchical structure. It is the result of architectural and urban design. Scene space is directly experienced but documented only implicitly and within itself. Unlike network space, information about scene space is hardly available over the Internet.

Table 3.1 Properties of network space and scene space.

Property	**Network Space**	**Scene Space**
Scale (Zubin)	type D	type D, composed of type C
Scale (Montello)	geographical	vista/environmental
Space type	map space	environmental space
Planning	pre-trip	on-trip
Process	selection	searching, following, exploring
Experience	mediated	direct
Time	absolute	relative
Structure	lines/nodes	scenes
Documentation	explicit	implicit

With respect to the domain of public passenger transport, network space comprises the network of lines and the timetable according to which these lines are operated. Scene space comprises the stops and interchange nodes, including their relation to the transport lines and their embedding into the surrounding environment. The bindings of public transport in line operation (Section 2.1) and the resulting elements of a journey (Figure 2.1) make an exposure of the traveller to both aspects of space unavoidable.

The separation of the public transport environment into a network space aspect and a scene space aspect is deeper than the scenario above might suggest. An analysis of the properties of the two types of spaces in the following section shows that it is a profound and cognitively significant distinction.

3.3 Properties and implications

Properties of network space and scene space are summarised in Table 3.1 and shall now be studied with respect to

- Lynch's "environmental image,"
- levels and types of space,
- human activities in space,
- interactions between network and scene space, and
- issues for modelling space.

Properties with respect to Lynch's "Environmental Image." Perception of an environment, network space or scene space, creates what Lynch [77] called an "environmental image," but its creation and the elements in this image differ for the two space types. Lynch writes (p. 8):

> "An environmental image may be analyzed into three parts: identity, structure, and meaning. [...] A workable image requires first the identification of an object, which implies its distinction from other things, its recognition as a separable entity. [...] Second, the image must include the spatial or pattern relation of the object to the observer and to other objects. Finally, this object must have some meaning for the observer, whether practical or emotional."

An "environmental image" can also result from a mediated and abstract presentation, like a network map; indeed, public transport networks can hardly be apprehended without an abstract presentation since they are too large, in the extreme they are spanning the entire global (airlines). It is not clear whether both direct and mediated experience result in equivalent environmental images. An experiment conducted by Lloyd et al. (1996) supports that information is encoded into the cognitive map in a perspective-free manner [75]. On the other hand, Presson et al. (1989) found evidence that information is encoded in a perspective-specific manner [99].

In either case, a working environmental image must contain identifiable *entities*. In network space, these are the nodes and the links between the nodes, but also lines and (departure) times. When travelling on a network, at each node the traveller has to decide which link to take next. Therefore, these nodes are also referred to as *decision points* in the literature (see, e.g., [61]). Identities in scene space are not so obvious. It is easy to enumerate some such entities (platforms, buildings, signs, etc.), but hardly possible to define them.

Structure is about how these entities relate to each other (and to the wayfinder). Again, for networks this is simple and obvious, because the network in itself is a well-defined structure, relating nodes to other nodes. This is especially true for networks in private transport. With public transport, this structure is more complicated and dynamic because of the binding to lines and the timetable (Section 2.1). In scene space, I propose the term *scene* to stand for a *local spatial configuration* of smaller entities contained within the scene, together with qualitative spatial relations among these entities.

As to the *meaning* of the environmental image, I follow the assumption that structure captures a lot of what an image means. For network space, this is especially evident: The network structure expresses connectivity, and

allows for decisions about route choice. In scene space, meaning has to be tied to scenes or parts of scenes. They express a local overview and can communicate *affordances* [36] like "here you may enter" or "there you may turn around." An analysis of affordances for wayfinding can be found in works by Raubal [100,104].

Levels of scale and types of spaces. Various classifications of space were reviewed at the beginning of this chapter. With respect to Zubin's (1989) four space types, both network space and scene space are type D spaces, that is, "territories," for they are beyond direct perception. However, scene space is composed of "scenes" and each scene classifies as a Zubin type C space. Typical scenes in a station environment are a platform, an underpass, a station hall, and a ticket office. Though they vary in size, they are all regions that cannot be apprehended at a single glance, but that can be *scanned* from a single vantage point, hence they are type C spaces. Zubin's type C spaces were my motivation for the name "scene space." Actually, we should call it "scenes space," because it generally consists of more than one scene, but the inconvenient pronunciation made me stick with "scene space."

In comparison to Montello's (1993) psychological spaces, public transport networks are on a geographical scale, that is, they are so large that they can hardly be apprehended without the help of symbolic representations, network maps in our case. Scene space is smaller, but still is much larger than the human body and always surrounds it. Individual scenes can be apprehended from a single place without significant locomotion, and hence qualify as vista spaces. An entire station, being composed of individual scenes, can no longer be apprehended without locomotion and thus is an instance of environmental space. Other than network space, the scene space constituted by a railway station or another node in the public transport system, can usually be apprehended without symbolic representations, given enough time to explore.

The classification by Freundschuh and Egenhofer (1997) makes the distinction between directly experienced and mediated spaces even more explicit by introducing the category "map space" that includes all symbolic representations of (large and small) spaces. These representations are subject to cartographic generalisation or, more generally, abstraction processes. This is clearly true of network maps: they are usually purely topological, and, hence, very abstract representations.

What none of the classifications reviewed addresses is the variability of network space over time. This topic will be touched upon a number of times in the remainder of this chapter.

Properties based on activities. From the designer's perspective (Section 2.1), both network space and scene space are explicitly documented in terms of plans, sketches, models, etc. These documents are indispensable tools for the process of the creation of networks and interchange nodes, a means of communication between builders, architects, engineers, that is, experts.

From the traveller's perspective, however, availability of information about network space and scene space is unbalanced. Documentation about network space is abundant: we have network maps and timetables, traditionally in printed form and more recently even on the Internet. Spatially relevant subsets are posted at stops and in interchange nodes. Temporally relevant subsets are even available by audible announcements. Information from these sources taken together describe the topology and the dynamics of a public transport service.

Information about the stops and interchange nodes (that is, about scene space) is sparse. The only information normally available is a list of facilities like catering and left luggage services, but such a list lacks spatial information. For example, there is usually no way to say where the luggage lockers are relative to some platform, without actually being (or having been) at the relevant station. Even though, sometimes, stations maps are put up, they are not available for pre-trip planning. Therefore, it is justified to claim that scene space is documented only implicitly.

This disproportionate availability of information about network space and scene space has an important consequence for travellers: *pre-trip* planning is only possible for the movement within network space; for scene space, *on-trip* planning is a necessity.[8] On-trip planning also includes considerations about auxiliary activities besides travelling, such as shopping while waiting for a connecting service.

The two space types influence human wayfinding means. In the mediated network space, wayfinding can be reduced to the process of selecting one or several connecting lines from the public transport network. This is mainly subject to the route choice behaviour theory as elaborated in [8]. It is different from travelling with a private car, where a sequence of route segments is chosen, largely independent of the four bindings mentioned in Section 2.1.

[8] Unless, of course, the relevant access, transfer, and egress nodes are already known from prior trips. But even then, if there is a problem in network space, like a disrupted service or a delay, scene space is where and on-trip is when these problems have to be compensated for.

In contrast, wayfinding in scene space is best described by terms like searching, following, exploring, and matching. There are no paths to choose from, since the environment constituted by interchange nodes is usually devoid of an evident network of paths. Rather, there are large spaces like halls and platforms, together with underpasses and corridors. In Allen's terms [3], people have to use piloting (between landmarks; in the example of Section 3.2, these were mostly signs) and oriented search (if an expected landmark cannot be seen).

Finally, time plays a different role in the two types of spaces. In network space, time has an absolute meaning in that it defines when trains and busses depart or arrive on an absolute time scale. In scene space, however, time is measured on a relative scale. In the example (Section 3.2), the connecting train departs at 09:06, which is a particular feature of the network space created by the Swiss railway system. The amount of time for the transfer is ten minutes.

This concludes the discussion of properties of network space and scene space. The sequence of elements of a journey using public transport (Figure 2.1) indicates that both spaces are necessarily involved in any trip, but there are other links between network space and scene space to be explored in the next few paragraphs.

Interactions between network space and scene space. Network space and scene space can be considered individually, but the application domain of public transport connects them in interesting ways:

1. Scene space ties the public transport lines,
2. Scene space penetrates network space, and
3. Network space controls behaviour in scene space.

These connections are now investigated in turn.

Scene space ties the public transport lines. The lines of public transport are tied together at the interchange nodes of the system and thus become what is described as a network. Without this "glue," the lines would be much like spaghetti in a plate. The distinction between linear elements (the lines) and linking elements (the nodes) is already present in Lynch's (1960) classification of spatial elements in the environmental image: paths and edges come together at nodes and regions.

In public transport, a third component is required to build a network: the timetable. If there is no connecting service, then the corresponding link in the network is essentially missing.[9]

Scene space penetrates network space. The traveller in public transport is always surrounded by scenes, even while sitting in a train coach, a tramway, or a bus. When there is a network, then it is an abstraction of what the traveller experiences. This experience can be (and is) reinforced by presenting the network in readily prepared maps and diagrams, which in turn makes scene space more navigable.

Network space controls behaviour in scene space. People navigate in stations mainly to reach some train or other transport vehicle. Since these vehicles arrive and depart at specific places at specific times, network space directly controls important behavioural parameters of people in scene space. In case of temporary problems such as delayed or cancelled services, this control is particularly evident. Travellers may suddenly find themselves with much more (or much less) time than they had anticipated. In this way, network space with its absolute notion of time as defined by the timetable induces the relative role time plays in scene space.

As far as public transport is concerned, network space and scene space are not two independent spaces, but two *aspects* of the same environment for wayfinding and spatial reasoning.

Modelling issues. It is tempting to use graphs to model environments for wayfinding. A graph is an abstract structure consisting of *nodes* and *edges* connecting some of the nodes. There are many different types of graphs and a discussion is beyond the scope of this chapter. Regardless of the type of graph that is being used, I posit that three questions must be asked and answered to ensure that the graph is a valid model:

1. What are the nodes? What is their cognitive significance?
2. What are the edges? What is their cognitive significance?
3. What nodes are connected and why?

For network space, answering these questions is straightforward because both, nodes and edges, are present in our everyday notion of public transport networks: nodes correspond to the interchanges and stops of the system, and edges correspond to the links between the stops and interchanges. The

[9] This is related to accessibility [71]. The lack of a measure of accessibility that takes this temporal component into account prompted a diploma thesis that develops such a measure [95,96].

typical network map of a public transport system, consisting of nodes and edges, reinforces our understanding of nodes and edges form a network.

Answering the same three questions for scene space is hard. We may try fitting a graph into the map of a station (or a university campus or a public park or a shopping mall). However, it is not at all clear where to place the nodes, nor how many nodes to place. Does a platform correspond to one node or to several nodes? How about a station hall or a station square?

There is an established modelling formalism for network space, namely graphs, but none for scene space. The remainder of this thesis is concerned with developing a modelling formalism for scene space. But just before turning to scene space, some modelling challenges that are specific to public transport networks shall be mentioned.

Network space can be formally represented using graphs. When modelling public transport networks, however, some complications arise that are not present in road networks: there are lines, and there is a timetable, but the network only results from integrating the lines over time. Note that lines and the timetable correspond to two of the four bindings presented in Section 2.1 and that both these bindings do not apply to private car traffic.

Further complications concern the lines that branch, contain circles (such as London Underground's "Circle Line") or take a different route on the direction from A to B than on the direction from B to A. All these problems can be modelled by adding attributes to the edges and nodes in the graph, as is done, for example, in a recent thesis about the optimisation of timetables in public transport [35] and in commercial timetable and network planning and querying systems (such as the German Hafas[10]), but in this case both the internal representation and the algorithms are proprietary.

But now on to developing a modelling formalism for scene space. Chapter 4 develops Schematic Geometry, a cognitively motivated yet formal representation for scene spaces, and Chapter 5 applies this formalism to represent the environment of a software agent that has to find a way from a given source to a given target in a railway station.

[10] `http://www.hacon.de/`

Chapter 4

Schematic Geometry

Gibson notes that "geometry began with the study of earth as abstracted by Euclid, not with the study of the axes of empty space as abstracted by Descartes" [36:132]. Common-sense conceptions of space are probably even farther away from Cartesian coordinates than Euclid's geometry. Section 2.2 indicated that human knowledge of space is not a coherent whole, but rather consists of largely independent fragments.

When investigating wayfinding in public transport, consideration of the fragmented and incoherent spatial knowledge of humans is a necessity. Moreover, it is vital to capture what the spatial configuration of an interchange node (such as a railway station) *means* to the traveller. To this end, a *Schematic Geometry* is developed, a formal model for scene space that builds on cognitive concepts. This chapter will

1. define cognitive spatial schemata for scene space,
2. define schematic geometry and its consistency rules,
3. introduce schematic geometry concepts, and
4. show how to implement schematic geometry.

4.1 Cognitive spatial schemata

When investigating railway stations, we observe that there are spatial elements that occur repeatedly. For example, most stations have a station hall, containing such things as timetables, waiting areas, doors connecting it to the station square, and platforms. Similar observations can be made for other scene spaces such as public parks, shopping malls, or airports. The elements identified can be abstracted as instances of image schemata, as introduced in Section 2.3. Stated the other way round, image schemata can be used to represent elements of scene space.

Image schemata are a useful basis for capturing the spatial semantics of scene spaces for these reasons:

1. they have instances that are located in space and that structure space;
2. they communicate a meaning and hint at potential activity; and
3. they can be combined to describe complex spatial configurations

This is very different from spatial models that build on points, lines, and polygons. These primitives are also located in space but have certainly no immediate meaning.

Image schemata are very versatile, but they miss some important details when considering human interaction with the environment they are supposed to represent. For example, the floor in a room is a SURFACE, but not *any* SURFACE is strong enough and horizontal enough for a human being to stand and walk on it. Similarly, a building is a CONTAINER, but not *any* CONTAINER is large enough for a human being to enter it. (Interestingly, precisely this distinction shows up in a design system for interactive fiction games; see [91] and Figure 2.3.) While image schemata describe the general structure and meaning of perception, they cannot handle aspects of usability that matter when considering an agent's actions in an environment.

These usability aspects are linked neither to the environment nor to the agent acting in it; rather, they are linked to the relation between agent and environment. This is precisely Gibson's concept of *affordance* [36]. The "enterability" of a CONTAINER is an instance of an affordance. By combining image schemata and affordances we can build more specific schemata. For example, we may require a CONTAINER to be enterable, thereby excluding a letter box, which is a perfect CONTAINER but not enterable for a human being.[11]

An investigation of local railway stations motivated the definition of six cognitive schemata that are essential for modelling scene space. These schemata are typeset in a bold sans-serif font so they can be easily distinguished from the underlying image schemata that are conventionally typeset in small capitals.

CONTAINER: a CONTAINER that is "enterable," affords support (is "stand-onable"), and is bounded; examples include a station hall and a waiting lounge.

[11] Raubal and Worboys note that affordances are often induced by image schemata [104:3.2.3]: for example, a CONTAINER affords enterability. While this is obviously true for a departure hall at an airport, it is not always true—a letter box is an equally obvious counter example. The problem with Raubal and Worboys' argumentation is that they attach affordances to image schemata (and thus features in the environment) and not to the *relation* between a (human) animal and its environment (as Gibson intended).

Figure 4.1 An unconscious link or ULINK connects two spaces without being consciously experienced when moving from one space to the other. In the image, the platform is connected to the small square (where the van stands) by means of a ULINK. Unconscious links have no physical manifestation.

REGION: a (soft-) bounded area or SURFACE affording support and perceived as a unit; for example, a "shopping area" or a "station square."

GATEWAY: a LINK affording "walk-through-ability" and which is consciously experienced when travelled through; for example, a door or an escalator.

ULINK: a LINK affording "walk-through-ability" in such an immediate and intuitive way that taking the link is unconscious to the wayfinder. See Figure 4.1 for an example.

AGGREGATE: a COLLECTION or set of things that belong spatially or functionally together, like a station that comprises a station square, a building, and a platform area.

OBJECT: a discrete entity in space (corresponding to the OBJECT image schema), a catchall for whatever might be relevant but none of the above; for example, a newspaper kiosk, a sign, or a timetable. (Depending on the purpose of the model, the newspaper kiosk could also be considered a CONTAINER.)

It is often convenient to abbreviate the schema names by their first letter: C, R, G, U, A, and O. For technical purposes it is also handy to have a special schema NONE, meaning that there is no schema instantiated. This only shows up in the implementation (Appendix C) but not in the discussion here.

Sometimes, more than one image schema is instantiated. This is called a *superimposition* [54:125] in the image schemata literature. An example for a superimposition was the newspaper kiosk above; depending on a traveller's intention, its perception will activate the CONTAINER or OBJECT image schema. For the present work it is important that

> any CONTAINER is an OBJECT and
> induces a COLLECTION of the objects it contains;

any SURFACE induces a COLLECTION of the objects located on it;
any PATH creates a LINK between its start and goal.

These relations indicate a classification of my cognitive spatial schemata into those having a "collecting" property, namely **CONTAINER**, **REGION**, and **AGGREGATE**, as opposed to those that do not. The schemata without a collecting property may be further classified into those with a "linking" property, namely **GATEWAY** and **ULINK**, and the remaining schemata, only **OBJECT** in this work. Table 4.1 summarises the six schemata along with their components, the relations they induce, and the classification.

Table 4.1 Cognitive spatial schemata

Schema	**Components**	**Relations**	**Class**
AGGREGATE:	none	part-of (element-of)	collect.[1]
CONTAINER:	inside, outside, boundary	part-of (contains)	collect.
REGION:	a bounded surface	part-of (on/off)	collect.
GATEWAY:	none	connect	linking
ULINK:	none	connect	linking
OBJECT:	the object	(not considered here)	other
PATH:[2]	source, dest.[3], trajectory	connect, left/right	linking

Notes: [1]collecting; [2]see main text; [3]destination.

Relations among schema instances. Schema instances do not occur in isolation. Rather, they are related to each other in a way that adds structure and meaning to the space in which they occur. For example (see Figure 4.2), a building instantiates the **CONTAINER** schema and thus can contain other instances, such as objects, doors connecting the inside and the outside (**GATEWAY**), and rooms (**CONTAINER** in **CONTAINER**).

In general, **CONTAINER**, **REGION**, and **AGGREGATE** all induce a *part-of* relationship, whereas **GATEWAY** and **ULINK** induce a *connection* relationship (Table 4.1). This can be directly derived from the nature of the underlying image schemata and the implicit superimpositions mentioned before.

Collecting schemata "span a space" in the sense of set-based geometry [136], that is, they have a significant spatial extent and the capability to contain other schema instances. Since the containment relation is transitive (if a contains b and b contains c, then a contains c), the structure that emerges from collecting schemata is a partonomy or, more formally, a partially ordered set (poset, see Appendix A). Whether this poset is strict or reflexive is largely a matter of taste. For the purpose of this thesis it is considered strict, that

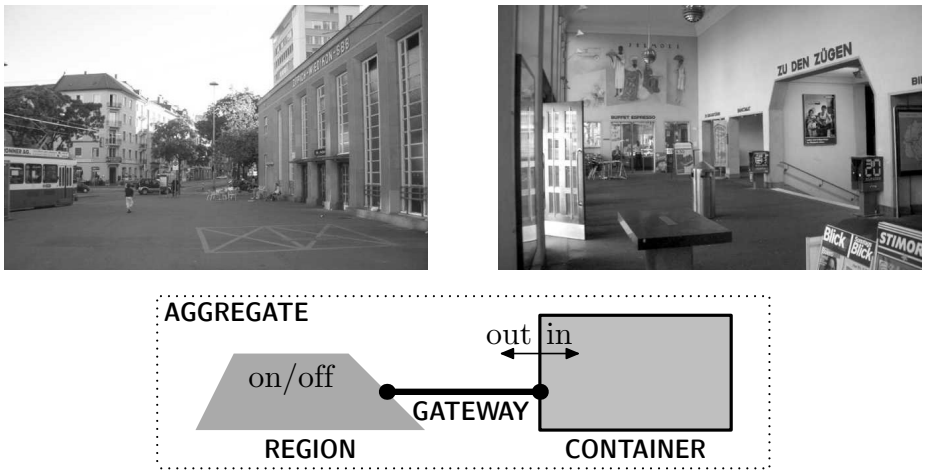


Figure 4.2 Relations among schema instances. The schema abstracts the photographs: There is a **REGION** (the square) and a **CONTAINER** (the building), both are linked by a **GATEWAY** (the doors), and all together constitute the station **AGGREGATE**.

is, a collection is never part of itself. More interesting is the question, if the poset is a lattice. This question will be investigated shortly.

Instances of the linking schemata (**GATEWAY** and **ULINK**) create connections between instances of the collecting schemata, and therefore between elements in the partonomy. The link itself can be considered yet another element in the partonomy, with two "parents," namely the two other elements it links together. Adding linking elements to the partonomy in this way results in an enlarged partonomy, but it does not invalidate any of the properties that must hold for a partially ordered set (as given in Appendix A).

Figure 4.3 shows the partonomy of schema instances, represented as a Hasse diagram (Appendix A), that arises from a simplified railway station. Figure 4.4 picks out a subset of the partonomy shown in Figure 4.3 and links it with a photograph of reality.

What about paths? PATH image schemata are very important in network space, but it turns out that there is no need for them when modelling railway stations. Note that a "path" in the sense of an enumeration (like "from the entrance, across the hall, down the escalator, into the shopping area") is referred to as a *route* and only metaphorically qualifies as a PATH because it lacks a physically manifest trajectory. For a street segment in network space, its trajectory is manifest in the form of a pavement, street markings, and so on.

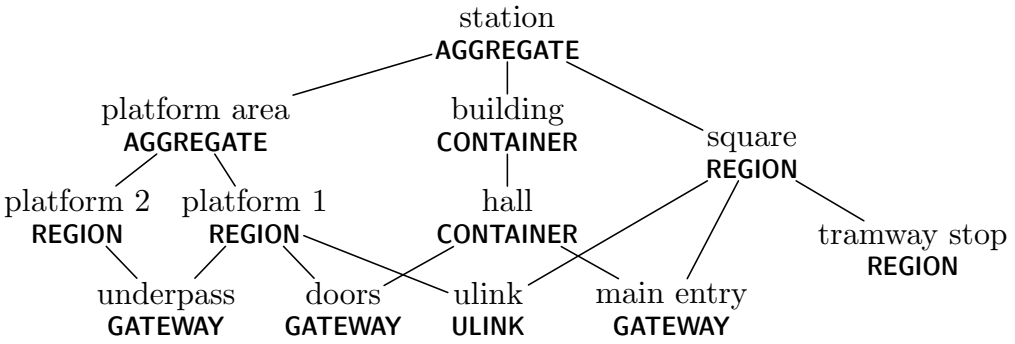


Figure 4.3 Schematic Geometry diagram for a simple railway station. Each element is identified by a short name such as "platform area" or "main entry" and its schema. The arrangement nicely shows the partonomy: the station comprises of a platform area, a building, and a station square; on the square is a tramway stop and the main entry leads into the building; etc.

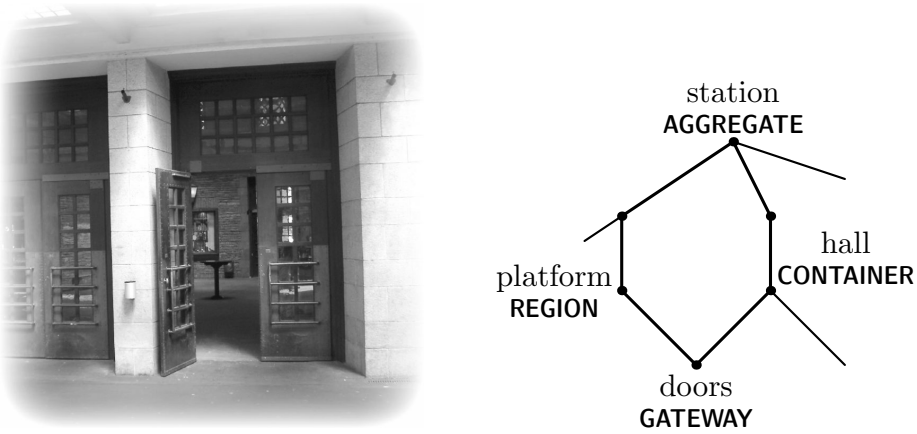


Figure 4.4 Schematic Geometry and the real world: a partial order expresses membership among instances of cognitive spatial schemata. The door on the photograph (a **GATEWAY**) links the hall (behind the door) and the platform (in front of the door). It belongs to both, the hall and the platform. By transitivity, it also belongs to the station as a whole, as do the hall and the platform.

GATEWAYS can be directed (like an escalator) or bi-directional (like a door). Since only the PATH image schema implies a direction but not the LINK image schema that underlies the **GATEWAY**, it is necessary to store a direction (or the fact that a link is bi-directional) explicitly in an attribute to the schema instance. Presently, all **GATEWAYS** are assumed to be bi-directional.

4.2 Defining Schematic Geometry

Building on the previous section it is now possible to define the Schematic Geometry as a partially ordered set of instances of the cognitive spatial schemata such that certain consistency rules are satisfied.

Definition. *A Schematic Geometry is a triple* (M, P, scm) *where* M *is a set of instances of the cognitive spatial schemata,* P *is a strict partial ordering relation, and* $\mathrm{scm} : M \to S$ *is a function assigning to each element in the schematic geometry the schema it instantiates;* $S = \{\mathsf{C}, \mathsf{R}, \mathsf{G}, \mathsf{U}, \mathsf{A}, \mathsf{O}\}$. *The arrangement of the schema instances in the partial order is governed by consistency rules defined below.*

The term "element" is used to refer to instances in M. If e in M and $\mathrm{scm}(e) = \mathsf{X}$, that is, if e instantiates schema **X**, then it will also be referred to as an **X**-element. Elements can have arbitrary (*key*, *value*) pairs as attributes. Attributes can be used to express, for example, that a room is only dimly lit, that a hall has a checkered floor, or that a sign (an instance of **OBJECT**) reads "Platform 3."

The ordering relation P represents membership or belonging-to in the sense of [1]. Therefore, if x and y are elements in M and (x, y) is in P, then x belongs to y. For the ordering to be meaningful, the following *consistency rules* C1 and C2 have to be satisfied. (Consult Appendix A for the meaning of poset-related terms like "minimal" or "greatest" or "lattice.")

C1: non-collecting elements are minimal (they can't contain)
C2: linking elements are not maximal (must be contained in what they link)

These rules are a direct consequence of the image schemata; the comment in parenthesis tells why they are required. Any violation results in a structure that is no longer meaningful, or at least confusing because normal human reasoning is disrupted. The example in Figure 4.3 satisfies these rules: the only non-collecting elements are the **GATEWAYS** and the **ULINK** and they are indeed minimal (C1); moreover, they are also not maximal, as required by C2.

Besides the consistency rules C1 and C2, it is often useful to require one or more of these optional rules:

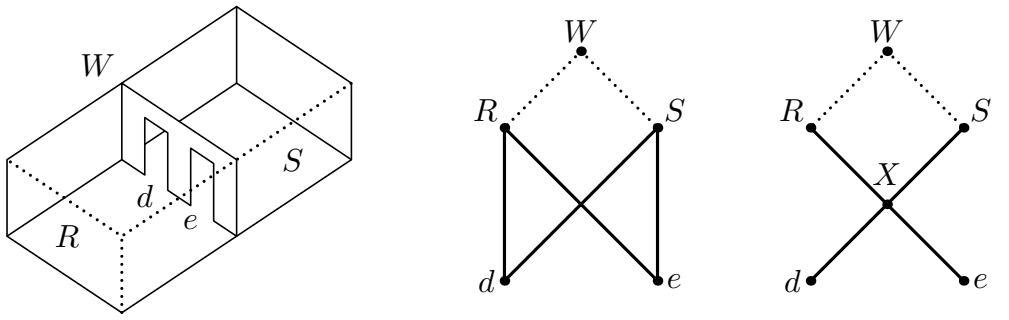


Figure 4.5 If in an environment W two rooms R and S are connected with each other by two doors d and e, then the resulting poset is not a lattice. However, it can be turned into a lattice by adding an "artificial" element, X. The interpretation of this new element is "some ⟨schema instances⟩," in this case "two doors."

C3: there is a greatest element, the study area
C4: non-linking elements have at most one immediate predecessor
C5: the poset (M, P) is an upper semi-lattice

These rules are optional because they are not required by the underlying image schemata. They can simplify representations and analyses at the price of sacrificing generality and are now discussed in turn.

Rule C3 is useful, for it states that there must be an element that corresponds to the entire study area. In Chapter 5, where software agents will navigate through Schematic Geometry models, there will always be a greatest element, namely "the station."

Rule C4 means that all elements have exactly one immediate predecessor, except for maximal elements, which have none, and linking elements, which have two (the underlying LINK image schema is a connection between two entities). This ensures that we have a tree if we remove all linking elements. Without this rule, collecting elements can also serve as links. Sometimes, this is desirable, for example to represent incomplete knowledge. However, whenever a collecting element serves as a link, then there must be truly linking elements that should be recorded in the model but are not. Therefore, this rule should be abided by during data collection.

Rule C5 would be useful because an upper semi-lattice is a more specific structure than an arbitrary poset. It also implies rule C3. Unfortunately, the simple situation of two rooms connected by two doors is a realistic counter-example (Figure 4.5). However, any poset can be turned into a lattice using a procedure known as *normal completion* [56]. This procedure adds "artificial" elements such that the semi-lattice condition holds. Other than in [58],

these new elements are not necessarily the intersection of existing elements. Rather, they represent a COLLECTION that is not explicitly modelled. In a sense, they represent plural forms such as "room A and room B are connected by *some doors*," and by going down in the semi-lattice, these "doors" are explicitly mentioned.

4.3 Schematic Geometry concepts

What can we do with Schematic Geometry? This section derives cognitively meaningful concepts that will be tested in Chapter 5, where they are put to work for simulating wayfinding.

Inheritance of location. An immediate consequence of the transitivity of the part-of relation is that elements in the partial order inherit the location of their predecessors, though this inherited location is less precise. For example, when waiting at a tramway stop that is located on a station square, then you are also standing on the station square. Simple as it seems, this is valuable information: it allows a traveller (or the software agent in Chapter 5) to widen its location awareness.

Dominance. The notion of "inheritance of location" can be more formally expressed with the concept of dominance, a concept from flowgraph analysis [20]. A flowgraph is a directed graph with a distinguished root node r from which all other nodes can be reached. If consistency rule C3 is assumed, the elements of a Schematic Geometry can be interpreted as the nodes of a flowgraph and the greatest element is the root node.

A node v is *dominated* by a node u if all paths from r to v contain u. The set of all dominators of a given node is linearly ordered. Consequently, each node v except the root has a unique immediate dominator, denoted $\operatorname{idom} v$. If a node v has exactly one immediate predecessor u, then $u = \operatorname{idom} v$. If it has more than one immediate predecessor, then it is no longer so trivial to determine its dominator, but there are several methods in the literature [20].

In terms of Schematic Geometry, the dominator of an element v is its "best predecessor," the element to which v undisputably belongs. This is especially interesting for instances of the linking elements, because they always have more than one immediate predecessor in the poset. In the usual Schematic Geometry diagrams (Figure 4.3, for example), going from v to $\operatorname{idom} v$ means "going up," to widen the spatial scope in a canonical way.

Scenes. Gateways are first-class objects in Schematic Geometry, because they closely correspond to image schemata. Scenes are different: there is no corresponding image schema and we only have the informal definition from Section 3.2. Building on Schematic Geometry, a scene can be defined relative to a given location as

> the smallest enclosing **CONTAINER**, or
> the largest enclosing **REGION**

if there is no enclosing **CONTAINER**. A scene is an instance of a **CONTAINER** if there is a clear boundary, for example, a station hall. A scene is an instance of a **REGION** if there is no such boundary, for example, a station square.

This description can be cast into a function scene : $M \to M$ operating on the elements of a Schematic Geometry (M, P, s), returning for each element in M its scene or the special value null if there is no scene: linking elements have no scene (they *link* scenes) and neither do **AGGREGATE** instances (they do not guarantee the coherence required by the definition of scene).

$$\text{scene}(e) = \begin{cases} \text{null}, & \text{if } \text{scm}(e) = \mathsf{A} \text{ (too large/incoherent to be a scene)} \\ \text{null}, & \text{if } \text{scm}(e) = \mathsf{G} \text{ (they connect scenes but have none)} \\ \text{null}, & \text{if } \text{scm}(e) = \mathsf{U} \text{ (same as for gateways)} \\ e, & \text{if } \text{scm}(e) = \mathsf{C} \text{ (smallest enclosing container)} \\ \text{regio}(e), & \text{if } \text{scm}(e) = \mathsf{R} \text{ (largest enclosing region)} \\ \text{idom}(e), & \text{if } \text{scm}(e) = \mathsf{O} \text{ (collecting schema that contains } e) \\ \text{null}, & \text{otherwise (to make scene idempotent)} \end{cases}$$

The scene function branches on the schema of the element whose scene is to be computed. The only complicated case is with **REGION** elements, where we have to "step up" the partonomy:

$$\text{regio}(e) = \begin{cases} \text{scene}(\text{idom}\, e), & \text{if } \mathit{canStepUp}(e) \\ e, & \text{otherwise} \end{cases}$$

where the predicate $\mathit{canStepUp}(e)$ means that

$$\text{scm}(\text{idom}\, e) \in \{\mathsf{C}, \mathsf{R}\} \text{ and } \forall x \in [\text{idom}\, e, e] : \text{scm}(x) = \mathsf{R}$$

and $[x, y] = \{\, z \mid x \prec z \prec y \,\}$ denotes a poset interval (Appendix A). "Stepping up" in the partonomy is only allowed if it does not interfere with the definition of scene from Section 3.2. This is the case if there are only **REGION** elements in the interval $[\text{idom}\, e, e]$ (see Figure 4.6). In my investigations of

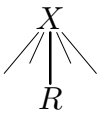
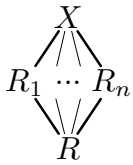
unique predecessor	several predecessors	
	shared subregion	the general case
		
$X = \mathrm{idom}(R)$ $[X, R] = \emptyset$	$X = \mathrm{idom}(R)$ $[X, R] = \{R_1 \ldots R_n\}$	$X = \mathrm{idom}^k(R)$ $[X, R] = W$
$\mathrm{scene}(R) = \begin{cases} \mathrm{scene}(X) & \text{if } \mathrm{scm}\, X \in \{\mathsf{R}, \mathsf{C}\} \text{ and } \mathit{canStepUp}(R) \\ R & \text{otherwise} \end{cases}$		

Figure 4.6 The scene of a region R. With a unique predecessor X that is a **REGION** or a **CONTAINER**, the scene of R is the scene of X, otherwise, R is its own scene. If there is more than one predecessor, we only "step up" if all are **REGION** instances. This is to ensure that the scene is not disturbed by intervening containers and their boundaries. In practice, such cases rarely occur. (R and the R_i are **REGION** instances, X is an instance of any schema, and W is the general interval $[X, R]$.)

railway stations, such complicated nested regions never occurred. Nevertheless, the definition has to cope with all theoretically possible configurations.

The scene function is idempotent: for all elements e, $\mathrm{scene}(\mathrm{scene}(e)) = \mathrm{scene}(e)$. Proof: For $\mathrm{scm}(e) \in \{\mathsf{A}, \mathsf{G}, \mathsf{U}, \mathsf{C}\}$ this follows immediately from the definition. For $\mathrm{scm}(e) = \mathsf{R}$, the scene function "steps up" as as long as *canStepUp* is true. When this process stops, scene returns its argument and is therefore trivially idempotent. Finally, for $\mathrm{scm}(e) = \mathsf{O}$, $\mathrm{idom}\, e$ is an instance of one of the collecting schemata, and these cases were already proved. □

Scenes can (and usually do) contain other elements, such as a station square to which belongs a tramway stop (**REGION**) and a newspaper kiosk (**CONTAINER**). Specifically, a scene s in M contains all those elements x in M that belong to s (that is, $x \prec s$ in P) but are not within a **CONTAINER** of the scene (containers have boundaries and thus limit the extent of a scene). This again gives rise to a function $\mathrm{grasp} : M \to \{M\}$ mapping a scene to the subset of M consisting of all elements that belong to the scene:

$$\mathrm{grasp}(s) = s \cup {\downarrow}s \setminus \bigcup\nolimits_{c \in C(s)} {\downarrow}c$$

where $C(s)$ is the set of all **CONTAINER** elements in $\downarrow s$.

By means of the functions *scene* and *grasp*, the informal notion of a scene from Section 3.2 is made explicit within the framework of Schematic Geometry. In the remainder of this text, however, "scene" will be used to mean both, the set of elements that belong to the scene as well as its representative topmost element. The *scene* function will be thought to return such an overloaded construct and context makes clear what is meant.

The scene graph. From a Schematic Geometry model of a scene space environment, a network called the *scene graph* emerges.[12] It is called the scene graph because it expresses connectivity between scenes. It is said to *emerge* because it gives us a network in an environment without an obvious network structure. The details of this "emergence" are now worked out.

When a traveller is located in a scene s and moves through a gateway g to another scene t, then this movement takes also place in all elements to which s, g, and t belong. However, the most specific description of this movement is on the level of the individual scenes and links. This observation motivates a classification of the "edges" (pairs $x \prec y$ in the Schematic Geometry (M, P, scm)) into "passable" edges and "abstract" edges. An edge is said to be *passable* if

> it has a **GATEWAY** or a **ULINK** as one of its endpoints, or
> it connects two **REGION** instances (the sub-region relation);

all other edges are said to be *abstract*. An edge is passable in the sense that actual locomotion in the environment corresponds to transitions along those edges, whereas actual locomotion only *implies* transitions along abstract edges. By convention, passable edges are drawn with solid lines and abstract edges with dotted lines. Using this convention, the scene graph becomes visually evident. Figure 4.7 shows an example.

This emerging network is probably the greatest benefit of the Schematic Geometry modelling approach: even though scene space is—by definiton—an environment without an obvious network structure, a network can be derived and used for all the typical network applications such as routing. Due to the image schematic foundation, this network is cognitively motivated and not merely imposed. It should be noted, however, that *all* locomotion takes place within the elements of a Schematic Geometry. Edges have no spatial extent, but merely express partonomic relation among the elements. This is a crucial difference from traditional network models, where the nodes are assumed to be dimensionless points and locomotion takes place along the edges.

[12] Not to be confused with the scene graph in computer graphics.

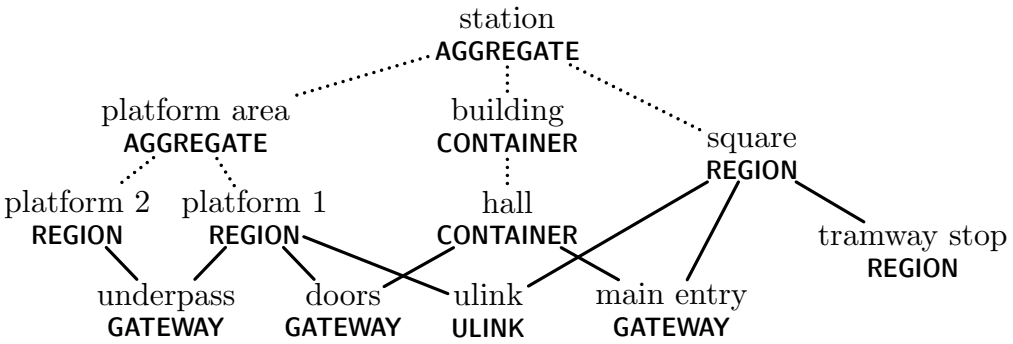


Figure 4.7 Classification of "edges" (pairs $x \prec y$ in the transitive reduction graph of the Schematic Geometry) as *passable* (solid line) or *abstract* (dotted line). This visualisation makes the *scene graph*, the "emergent network," evident.

Finally, the gateways and unconscious links give rise to a function "through" that represents the action of going through a link from one scene to another scene. Given an instance of a GATEWAY or a ULINK, and an element at either side, it returns the least element at the other side.

Sketch visualisation. The diagrams used so far to visualise a Schematic Geometry nicely show the partonomy and the emergent network. However, they give no hint as to the real (geo)metrical layout. Though all metrical details were consciously left out of the model, it can be useful to include them every once in a while. This desire leads to an alternative "sketch" visualisation that is closer to a plan used in architecture, but typically not to scale. Figure 4.8 shows a sketch visualisation of a Schematic Geometry.

Operations. Many of the concepts presented in this section can be formulated as operations on the elements in a Schematic Geometry. Collecting them here nicely summarises this section:

scm : $M \to S : x \mapsto$ the schema that x instantiates

idom : $M \to M : x \mapsto$ the immediate dominator of x

scene : $M \to M : x \mapsto$ the scene of x

grasp : $M \to \{M\} : s \mapsto$ the "contents" of scene s

through : $M \times M \to M$: given a linking element and a reference element at either side of the link, find the link's immediate ancestor on the other side

This is also a good place to remember (page 53) that "scene" means both, the result of scene(x) as well as the result of grasp(scene(x)), and context will clarify. Consequently, the composition grasp(scene(x)) will be, incorrectly

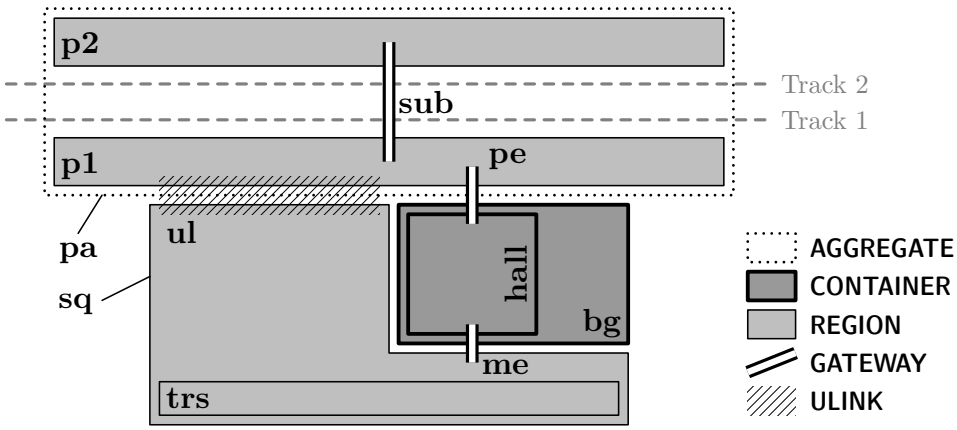


Figure 4.8 Sketch visualisation of the Schematic Geometry from Figure 4.3 along with the legend. The nesting of polygons expresses the partonomy. Gateways and Ulinks are drawn such that they start and end in the two elements that they link. Note that these sketches are only useful if the Schematic Geometry to be visualised adheres to consistency rule C4.

but conveniently, abbreviated as scene(x). For example, Table 5.1 (page 64) says "$t \in$ scene(p)" to mean "$t \in$ grasp(scene(p))," what is clear by the presence of the $\in$ symbol.

4.4 Implementing Schematic Geometry

Implementing Schematic Geometry means encoding the partial ordering of the schema instances in such a way that the operations from the previous section can be easily performed. The attributes of an element in a Schematic Geometry can be conveniently stored in any implementation of the dictionary abstract data type. The schema scm(e) of an element e can be treated as an attribute with a special name that must not be used for ordinary attributes. The only challenge lies in finding a suitable encoding for the poset.

A search for literature about poset encoding revealed that there is no ready solution that handles all the desired operations (order test, cover test, suprema, minima, maxima, and bounds). This was surprising, as posets frequently show up, both within computer science (think of class hierarchies) and more application-oriented domains such as scheduling in manufacturing or even in the study of ontologies [55]. Basically, there are three approaches to representing posets, each with specific advantages and problems:

1. transitive reduction graph
2. bit-vector encoding
3. realiser (set of linear extensions)

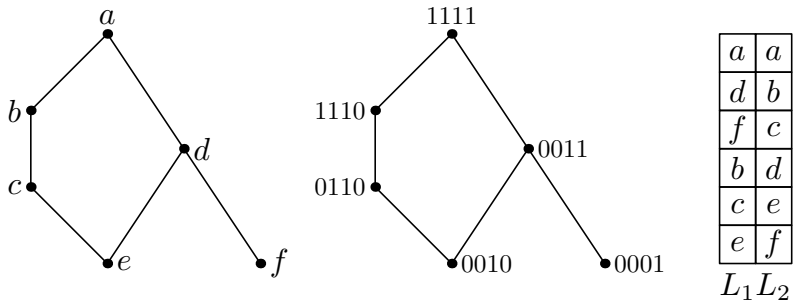


Figure 4.9 Illustration of the three approaches for encoding a poset: transitive reduction graph (left), bit-vectors (middle), and realiser (right); see text.

All approaches are sketched in Figure 4.9 and briefly explained here. Approach (3) is fully developed in Appendix B.

Storing the transitive reduction graph of a poset is the most obvious approach, since it corresponds with the usual way that posets are visualised. The problem with this approach is that a basic order test (test if $x \prec y$) requires searching for a path from x to y in the graph. On the other hand, computing covers and testing the cover relation is straightforward and incurs low computational costs.

The idea of the bit-vector approach is to assign each element x of the poset a bit-vector $\varphi(x)$ such that $x \prec y$ iff $\varphi(x) \subset \varphi(y)$. The challenge is to find a set of vectors with as few components as possible [28]. A special case of bit-vector encoding is the embedding of a poset into an n-dimensional cube:[13] the set of all subsets of an n-element set ordered by the subset relation. This is a lattice. If the poset to be embedded is also a lattice, then computing infima and suprema corresponds to intersection and union of bit-vectors.

The last method to be considered uses a realiser to encode a poset. A realiser is a set of linear extensions of a poset such that the intersection of those extensions results in the original poset [126]. Linear extensions lend themselves well to computer representation, because they can be easily stored in arrays. Moreover, it is rather easy to compute a number of poset operations. Most importantly, a poset element x precedes an element y if (and only if) x precedes y in each linear extension of the realiser. The difficult part is to find a method to compute a good realiser. Appendix B develops such a method.

[13] Thanks to J. Nievergelt for suggesting this method.

Chapter 5

Simulating Wayfinding

In this chapter a software agent is developed that performs wayfinding in a schematic geometry model of a railway station. The task is to find a way from a given start to a given destination. The only information available to the agent is the structural information about a station as encoded in a Schematic Geometry model.[14] Note that this wayfinding simulation involves two models: the Schematic Geometry as a model of the environment (scene space), and the agent, which is itself a model that operates within the Schematic Geometry model. Both models were implemented in a computer program called "Odeon" that served as proof of concept and was used to perform the actual experiments. See Appendix C for information about Odeon.

5.1 Software Agents

According to Russel and Norvig's "Intelligent Agent Book," [111:31] an agent is anything that can be viewed as *perceiving* its environment through *sensors* and *acting* upon that environment through *effectors* (Figure 5.1). A software agent is a program that is run by a machine and implements the mapping from percepts to actions. Designing such agent programs is a traditional artificial intelligence job [111:35].

Designing an agent needs clarity about the possible percepts and actions. They certainly depend on the environment, in which the agent "lives." If the agent is supposed to achieve a goal, then this goal must also be specified. This requires four pieces of information to be known when designing an agent: the percepts, the actions, the goal(s), and the environment. They are collectively and conveniently referred to as an agent's PAGE description [111:36].

[14] Computational modelling is a good exercise in not using information that is not supposed to be used.

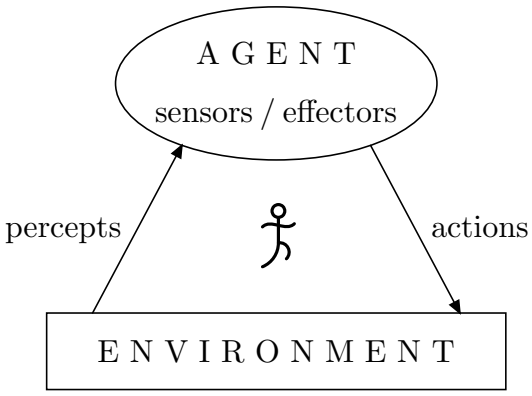


Figure 5.1 An agent is anything that interacts with an environment through sensors and effectors (after [111:32]).

Agents also have *internal state* information. For a goal-directed wayfinding agent, this minimally includes the destination so that it can be recognised when it is reached. More elaborate wayfinding agents may also construct/use a cognitive map of the environment, thereby learning the environment (Section 2.2). The considerations so far can be condensed into the following skeleton for a generic software agent [111:38]:

function agent(*percepts*) **returns** *action*
  static *state* # *agent's internal state*
  state := update(*state*, *percept*)
  action := choose-best-action(*state*)
  state := update(*state*, *action*)
  return *action*

Internal state is a function of past percepts and chosen actions. It is consulted to decide on the next action. The "static" keyword indicates that the *state* is preserved between invocations of the function.

My wayfinding agent lives in a scene space, represented as a Schematic Geometry (M, P, scm), and will be given the task to go from an initial place to a destination place, maintains only minimal state, and closely follows the generic agent skeleton shown above. Its PAGE description reads as follows:

Goal: find a way to a given destination.
Environment: scene space, represented using Schematic Geometry.
Percepts: the current scene.

There are two parametrised actions that are essential to the system:

goto ⟨destination⟩ (when ⟨destination⟩ in current scene) and
through ⟨link⟩ (where ⟨link⟩ belongs to the current scene).

Two more actions exist, but they are merely technicalities:

up (performed as part of perceiving the environment when the agent notices that it is not at the top of the current scene; it corresponds to perceiving the entire scene even though the agent is at a specific location within that scene), and
done (the destination was reached).

The destination is considered as reached if the present location *belongs to* the destination, not necessarily if it *equals* the destination. This is a reasonable choice in a hierarchically structured environment. For example, if the destination is the station building, it will be considered as reached once the agent is inside the building.

The resulting agent program looks like this:

function wayfinder(*currentElement, currentScene*) **returns** *action*
  static *source, target*: Element
    memory: Set of Element
    route: List of Element

  assert *currentElement* ∈ *currentScene* # *safety*

  # *Update internal state*
  append *currentElement* to *route*
  add *currentScene* to *memory*

  # *See if we reached the target, else go up*
  if *currentElement* ⪯ *target* **: return done**
  if *currentElement* ≠ *currentScene* **: return up**

  # *Target in current scene, i.e., visible and reachable?*
  if *target* ∈ *currentScene* **: return goto** *target*

  # *Target not in current scene, so we have to leave it!*
  set *link* := chooseLink(*currentScene, route, memory*)
  if *link* = null **:** set *link* := last link from *route*
  if *link* = null **: return done** # *trapped!*
  append *link* to *route*
  return through *link*

The assertion makes sure that the wayfinder is invoked with correct arguments. Next, internal state is updated by appending the *currentElement* to the *route* and adding all elements of the *currentScene* to the agent's *memory*. Then the simple cases are handled: target recognition, perception of the whole scene (going "up"), and realising that the target is in the current scene (resulting in a **goto** action). Finally, if there is no simple case, the chooseLink function handles the non-trivial decision-making. This is classical symbol processing and will be explained in the next section, after a quick note about embodiment and situatedness.

This wayfinding agent is *situated* [18,97:72]: it is on its own in the environment and can only rely on the percepts from his sensors; there is no input from an operator. The agent, being realised entirely in software, is not *embodied*. If "new AI" is equated with embodied cognitive science (as is done in [97]), then the agent clearly belongs into the framework of classical AI. However, the image schemata that underly Schematic Geometry are an intrinsically embodied concept (Section 2.3); if Schematic Geometry is interpreted as part of the agent's cognitive system, instead of its environment, then the agent is embodied.

5.2 Standard Wayfinder

This agent is called the "standard" wayfinder to distinguish it from two other agents to be developed in Section 5.3. The standard wayfinder follows Wiener's fine-to-coarse planning hypothesis [134], a cognitive model that describes a possible use of hierarchically structured information for wayfinding and assumes that there is no coarse (and complete) plan of a route. The idea is to use only minimal previous knowledge *about* the environment but use all the structural information *in* the environment; in Donald Norman's terms: the standard wayfinder has minmal *knowledge in the head* and relies maximally on *knowledge in the world* [93].

So how can the agent exploit the information that is present in a Schematic Geometry representation of the environment? The scene is what is both immediately reachable and available to perception. If the destination is in the current scene, then reaching it is a trivial **goto** ⟨destination⟩ action. For what is beyond the current scene, the agent uses these rules and heuristics to choose a good **through** ⟨link⟩ action:

– Simple cases like dead-ends or "tunnels" (scenes with only two links) are handled according to the rule "don't go back" (unless there is no other option).

– Previews, like glass doors, that allow the agent to "see" what is beyond a GATEWAY is another valuable source of information.
– Rumelhart's Room Theory, re-interpreted for wayfinding, provides two very useful heuristics for wayfinding: two rules to prefer presumably "good" linking elements in the current scene over others, which seem less promising (see below).

The agent also maintains minimal state information: the source $s \in M$, the destination $t \in M$, the route travelled so far (a sequence of elements $x_i \in M$), and a configurable amount of previous knowledge in terms of a set $K \subseteq M$ of elements that are known (i.e., can be recognised) by the agent. This set of known elements grows as the agent travels around and is used for deciding if previews or the Room Theory heuristics are applicable.

Rumelhart's Room Theory. In 1974, David E. Rumelhart devised a method for computing context-sensitive answers to *where* queries: the least precise region that contains the target place but excludes the reference location and is contained within the "room," which is the least region which contains both the target and the reference [110], reported in [64]. For example, when asking someone in Europe where the Empire State Building is, the answer is expected to be "in the United States," but when asking the same question while in New York, the answer is probably "on 34th street." Figure 5.2 shows another example.

The Room Theory was criticised based on an empirical study [113] that revealed answer patterns that do not conform with the Room Theory. For example, humans often give answers on the town level, so even someone in Europe would say that the Empire State Building is in New York, not in the United States. Of course, the answers predicted by the Room Theory very much depend on the database of nested regions, upon which the theory operates.

Irrespective of these critiques, the Room Theory provides two very useful wayfinding heuristics—it only needs to be translated into Schematic Geometry. This translation is easy: each element of a Schematic Geometry corresponds to a "region" in the Room Theory, the room of the current scene s and a destination element t is the least upper bound of s and t, and the room-referent, Rumelhart's term for the context-sensitive answer to a *where* query, is the one element short of the room when stepping up from t by repeatedly setting $t := \mathrm{idom}(t)$. If the wayfinder's current scene is taken as the reference location and the wayfinder's destination as the place of inquiry, then the two heuristics are as follows:

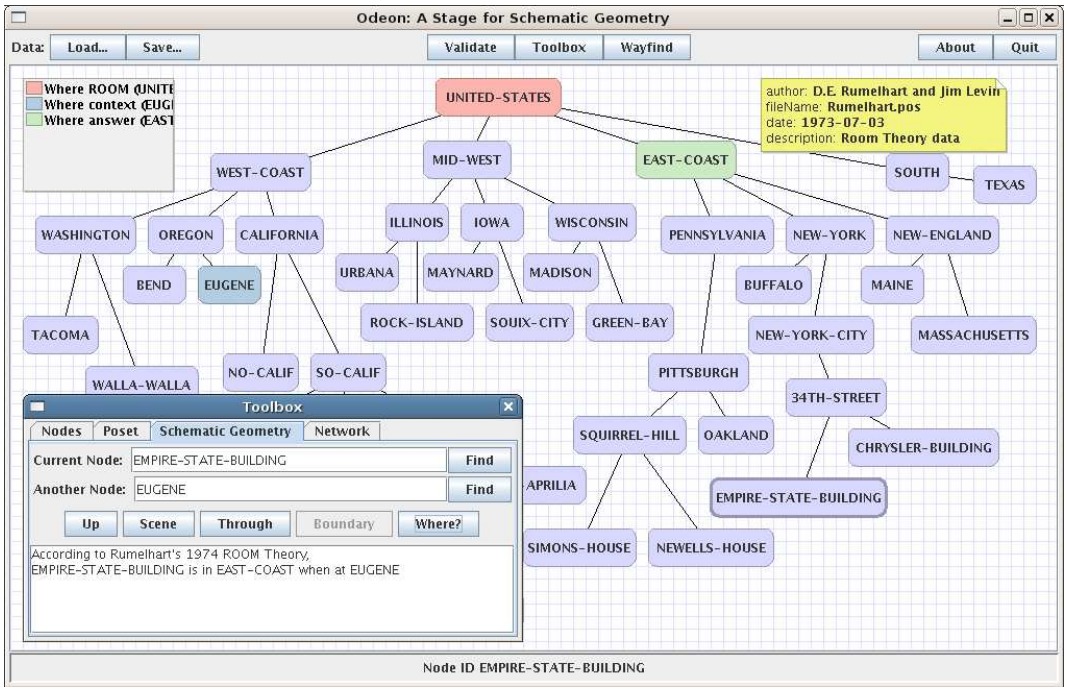


Figure 5.2 Rumelhart's Room Theory [110]: The Odeon user interface shows that "East Coast" is the context-sensitive answer to the query "Where is the Empire State Building?" when the context is "Eugene."

1. Do not leave the room (*stayInRoom*). Leaving the room inevitably takes the wayfinder farther away from the destination. This heuristic fails if there are places within the current room that are connected with each other only through places outside the current room.
2. Take a link that leads into the room-referent (*focusSearch*). This heuristic takes the wayfinder closer to the destination. It can only be applied if there are links from the current scene into the room-referent.

The standard wayfinder implements both heuristics. Whether they are actually used can be controlled through the options *stayInRoom* and *focusSearch*. If both are turned on, the agent uses them as follows.

At each scene which does not contain the destination (that is, whenever a **through** ⟨link⟩ action is required), the agent determines all links in the current scene, the current room, and the room-referent. The latter two can only be determined if they correspond to elements in the Schematic Geometry that are known to the wayfinder (previous knowledge or part of the current scene). A useful feature of the Room Theory heuristics is that both the room and the room-referent tend to be elements high up in the Schematic

Geometry, that is, elements that likely belong to the wayfinder's previous or common-sense knowledge.

Once the room and the room-referent are known, the agent partitions the links of the current scene into the set *badLinks*, those leading out of the room, and the set *goodLinks*, those leading into the room-referent.[15] If there are good links ($goodLinks \neq \emptyset$), the agent will choose one of those. If there are none, it will at least avoid taking bad links, unless, of course, *badLinks* comprises all links in the current scene.

Using previews. Previews show what is beyond a gateway. They support the agent in choosing a particular gateway out of several gateways. Whether or not there are previews is determined by the architecture. Glass doors, for example, usually offer a preview of the space behind the doors. Previews are conceptually the same as pointing signs, like "this way to the platforms."[16]

Previews are not objects in Schematic Geometry but can be represented as an attribute of an edge (x, g) where g is the gateway and x an ancestor in the same scene as g, typically the scene's top. The attribute's name is "preview" (by convention) and its value is a comma-separated list of the elements of the Schematic Geometry that are previewable through g.

Table 5.1 lists the situations that can be exploited for wayfinding. It is ordered from most specific to least specific: given a preview p and a destination t, the table should be searched in order for a matching situation. If no situation matches, then the agent cannot use this preview to make a decision. Note that a situation only matches if the agent can recognise the preview and, if relevant, its context, that is, if they are part of the agent's accumulated knowledge K. The only exception is the first case in the table ($p = t$), because we always assume that the agent recognises its destination t.

[15] These sets are always disjoint. Proof by contradiction: Let r be the room and f the room-referent for a given current location and a given destination. By definition of room-referent and room, $f \prec r$ in the Schematic Geometry. Assume x is in both *goodLinks* and *badLinks*: this means that $x \preceq f \prec r$ and $x \not\preceq r$, a contradiction. □

[16] I suggest that there are two types of signs, those that identify an object and those that point towards an object. Identifying signs help with goal recognition and pointing signs help with finding the destination. Note that even the pointing signs are identifiers, namely by identifying the direction or door or path that leads towards the destination. Pointing signs visually appear as arrows or in an arrow-like manner. Page 86 resumes this discussion.

Table 5.1 Situations where previews can be used

Situation	Description
$p = t$	The destination can be seen by means of the preview: there is no stronger argument for taking that gateway.
$p \prec t$ and $p, t \in K$	The preview belongs to the destination and both, preview and destination, are known. Once through the gateway, the destination will be reached.
$t \in \text{scene}(p) \in K$ $p \in \text{scene}(t) \in K$	Both p and t are in the same scene. This implies unimpeded movement between p and t, so going to preview p must take us to the destination.
$t \prec p$ and $p \in K$	The destination belongs to the preview, but is not necessarily in the same scene. Taking the corresponding gateway is a good heuristic but will fail if the gateway leads to a scene that is not directly connected with the destination's scene.

Symbols: $p \in M$ a preview, $t \in M$ the destination, $K \subseteq M$ the agent's current knowledge as the set of known elements from the Schematic Geometry (M, P, scm).

Using salience. The salience of gateways can be used as a weight to increase the probability that a gateway is chosen. If the current scene has the gateways $g_1, g_2, \ldots g_n$ and $s(g_i)$ is the salience of gateway g_i, then the probability of gateway g_i being chosen as the one through which to leave the current scene is $s(g_i)/\sum_{j=1}^{n} s(g_j)$. The use of salience is controlled by the *useSalience* option. If salience is not used, then all gateways have the same probability of being chosen. As with previews, salience is an auxiliary concept in Schematic Geometry and therefore stored as an attribute.

Implementation. The generic scene space wayfinder agent (page 60) invokes the function "chooseLink" to determine the best link if the destination is not in the current scene. ChooseLink implements the heuristics and rules just described and is shown here:

```
function chooseLink(currentScene, route, memory) returns element
  local curloc: Element   # current location
        links: Set of Element   # choice set
        room, room-referent: Element
        goodLinks, badLinks: Set of Element

  ⟨Put all links in currentScene into links⟩
  set curloc := last element of route
```

if *usePreviews* **:** ⟨Check previews according to Table 5.1⟩
  if ⟨Matching *link* found using previews⟩**: return** *link*

⟨Compute the *room* and the *room-referent*⟩

if *stayInRoom* **and** $room \in memory$ **:**
  set $badLinks := \{\, x \in links \mid \text{through}(x, curloc) \not\preceq room \,\}$
  if $|links| > |badLinks|$**:** ⟨Remove all *badLinks* from *links*⟩

if *focusSearch* **and** $room\text{-}referent \in memory$ **:**
  set $goodLinks := \{\, x \in links \mid \text{through}(x, curloc) \preceq room\text{-}referent \,\}$
  if $goodLinks \neq \emptyset$**:** set $links := goodLinks$

if *dontGoBack* **and** $|links| > 1$ **:**
  set *lastLink* := last linking element from *route*
  ⟨Remove *lastLink* from *links*⟩

return pickLinkFromList(*links*)

Informal descriptions in angle brackets ⟨...⟩ represent distracting implementation details.

After collecting all links (GATEWAYS and ULINKS) from the current scene (human equivalent: scanning the scene) and determining the current location (which is always the last element in the route), the agent checks if there is a helpful preview. Previews are checked before the Room Theory heuristics are applied because previews either lead directly to the destination, or are equivalent to the *focusSearch* Room Theory heuristics (the last case in Table 5.1).

If there are no suitable previews (or if the agent has been configured not to use previews), the two Room Theory heuristics are used to reduce the choice set, which initially comprises all links in the current scene, as much as possible. Note that these heuristics can only be applied if the room or the room-referent are known, otherwise they would be meaningless to the agent.

Finally, the "pickLinkFromList" function is invoked to choose a link from this reduced choice set, honouring the agent's *useSalience* parameter as described previously.

5.3 Perfect and Random Wayfinders

For evaluation purposes two other wayfinding agents are built, a perfect wayfinder and a random wayfinder. Both navigate on the emergent network consisting of passable edges only (Section 4.3).

The **perfect wayfinder** always finds a shortest route: there can be no shorter routes than the one travelled by a perfect wayfinder, but there may be several equally short routes. This agent is implemented using Dijkstra's single source shortest path algorithm [24]. There is no cognitive evidence for this method, but it is used only as a benchmark with which to compare the standard wayfinder.

The **random wayfinder** repeatedly determines all links in its current scene and then chooses one at random. This process continues until it is cancelled or the destination is reached (by chance). This agent makes no use of information in the environment and its behaviour corresponds to random walking. Again, this is useful as a benchmark with which to compare the standard wayfinder.

5.4 Study stations: Enge and Wiedikon

The StandardWayfinder shall be tested in the Schematic Geometry representations of two railway stations in the city of Zürich: Enge and Wiedikon.

Enge. The railway station "Zürich Enge" is one of the small stations in the city of Zürich, an access point to the rapid transit system for the "Enge" district, and a stop for four tramway lines. It was built in 1925–1927 by the brothers Otto and Werner Pfister after the example of the new main station in Stuttgart (Paul Bonatz and Friedrich E. Scholer, 1912–1922). It is characterised by its plain granite façade and an array of columns facing the station square (called "Tessinerplatz"), giving the building a monumental appearance [76].

The station's schematic structure is shown in Figure 5.3. It is the classical structure with a station building, a station square, and a platform area. Somewhat non-conventional is the secondary tramway stop to the side of the station building. There is a very convenient connection from this tramway stop directly to platform 1. It is not very salient and therefore not expected to be used by the casual traveller. For our structural investigation, however, this lacking salience of an otherwise useful connection is of no concern. There is no line of sight between the main tramway stop on the station square and the additional tramway stop to the side.

The Enge Railway Station

station square (**sq**)

cr and **co**

platforms (**pa**)

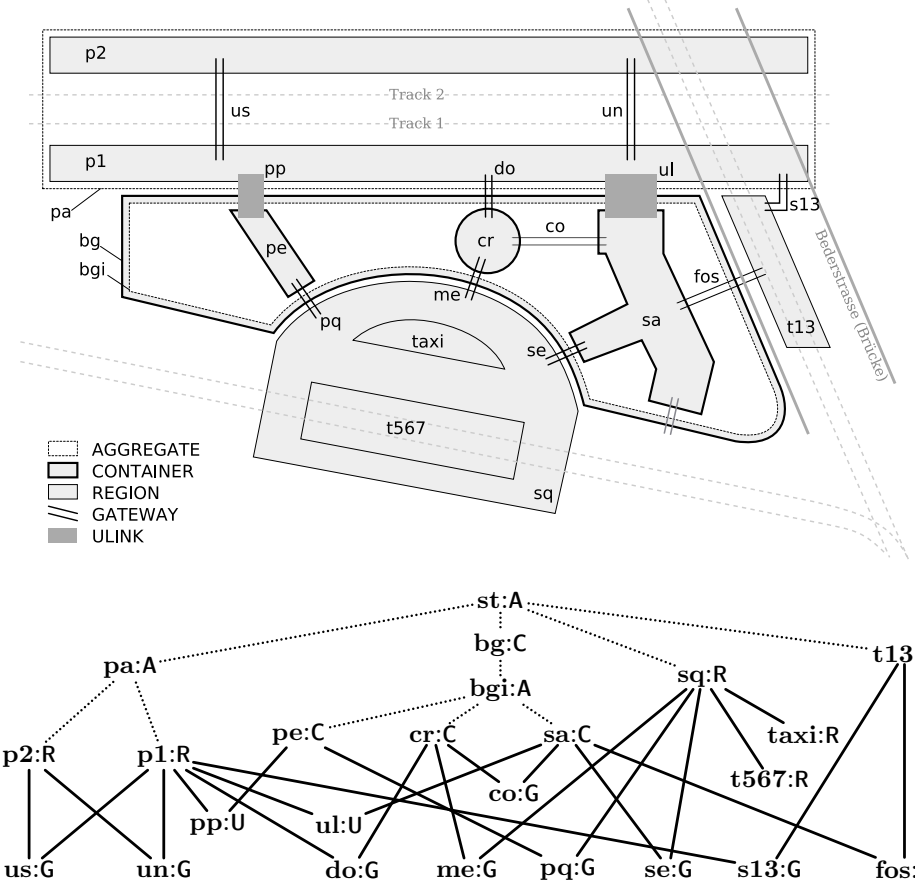


Figure 5.3 Schematic Geometry for the Enge railway station. Elements: **bg**:C Building, **bgi**:A Interior (of building), **co**:G Corridor, **cr**:C Circular Room, **do**:G Doors (glass), **fos**:G Flight of Stairs, **me**:G Main Entry, **p1**:R Platform 1, **p2**:R Platform 2, **pa**:A Platform Area, **pe**:C Passage, **pp**:U Ulink p1/pe, **pq**:G Side Entry sq/pe, **s13**:G Stairs t13/p1, **sa**:R Shopping Area, **se**:G Side Entry sq/sa, **sq**:R Station Square, **st**:A Station, **t13**:R Tramway Stop (line 13), **t567**:R Tramway Stop (lines 5, 6, 7), **taxi**:R Taxi Stand, **ul**:U Ulink p1/sa, **un**:G Underpass North, **us**:G Underpass South. For convenience, this figure also appears on a foldout at the end of the text.

The Wiedikon Railway Station

sq and **bg** with **me**

the hall

pa (seen from **s2**)

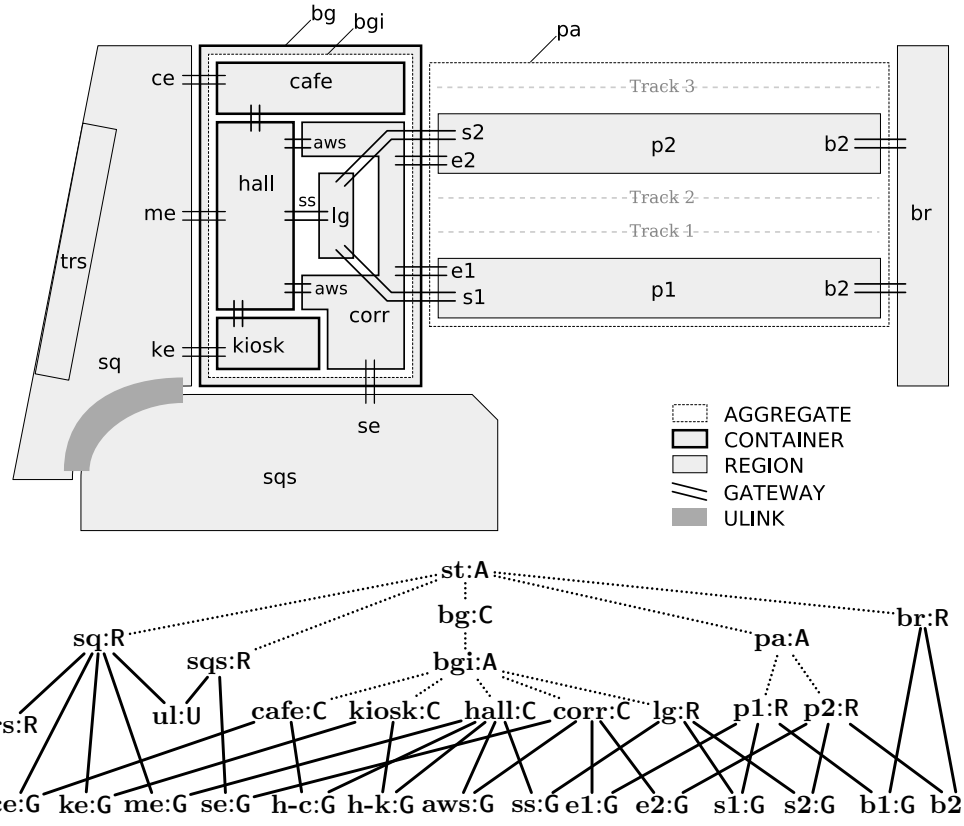


Figure 5.4 Schematic Geometry for the Wiedikon railway station. Elements: **aws**:G Two Archways, **bg**:C Building, **bgi**:A Interior (of building), **b1**:G Stairs br/p1, **b2**:G Stairs br/p2, **br**:R Bridge, **cafe**:C Café, **ce**:G Entry sq/cafe, **corr**:C Corridor, **e1**:G Elevator corr/p1, **e2**:G Elevator corr/p2, **h-c**:G Door hall/café, **h-k**:G Door hall/kiosk, **hall**:C Hall, **ke**:G Entry sq/kiosk, **kiosk**:C Kiosk, **lg**:R Landing, **me**:G Main Entry, **p1**:R Platform 1, **p2**:R Platform 2, **pa**:A Platform Area, **s1**:G Stairs lg/p1, **s2**:G Stairs lg/p2, **se**:G Side Entry, **sq**:R Station Square, **sqs**:R Side Square, **ss**:G Stairs hall/lg, **st**:A Station, **trs**:R Tramway Stop (lines 9 and 14), **ul**:U Ulink sq/sqs. For convenience, this figure also appears on a foldout at the end of the text.

Wiedikon. The railway station "Zürich Wiedikon" is the only "Reiterbahnhof" in Switzerland, that is, a type of station where the building is atop the tracks and platforms, which are accessed through stairs and elevators going down from the building. It links the rapid transit system with two tramway lines, several bus lines, and the "Aussersihl" and "Wiedikon" districts of the city. The station building houses a two-storey station hall (which still boasts promotional paintings from the 1930ies), a café, and a kiosk. It was built in 1926 by Hermann Herter and recently has been renovated (1993–1998).

The station's schematic structure is shown in Figure 5.4. It follows mostly the prototypical pattern of a station building, a station square, and a platform area, though the station square is actually split up into two parts, a main square in front of the building where the main entrance is and the tramways stop, and a side square, where the busses stop. There is no crisp boundary between the two, giving rise to an unconscious link. The bridge (br), though independent of the station, serves as an overpass and therefore is considered part of the station.

During the renovation, a "line of sight" was created by making two windows in the interior of the building between the corridor and the landing. The line of sight is cafe-corr-lg-corr-se. Unfortunately, there is no information available about the effect of this line-of-sight on wayfinding or other human activities.

In contrast to the Enge station, all building parts are linked with one another such that it is possible to visit them all without leaving the building. Another structural difference is that the two platforms are equally accessible; a typical feature of "Reiterbahnhof" types of stations. A final structural difference is that the Enge station offered a convenient shortcut directly from the secondary tramway stop (**t13**) to platform 1 without having to go through the building; at the Wiedikon station, all access to the platforms is through the building.

The data was collected by looking for schema instances *in situ*:

1. Identify all gateways (list them at the bottom of the page)
2. Identify the elements they link (list them above the gateways)
3. Group these elements into collecting schema instances
4. Recursively group these collecting elements as appropriate
5. Look for regions that are connected by unconscious links

Finally, gateways were examined for previews they offer.

5.5 Simulation

The agents and their environments are now ready for simulation. The goal of this simulation is to find out how well the standard wayfinder performs in various tasks. Its performance is measured by the length of the route travelled, compared to the length of a shortest route (as computed by the perfect wayfinder). The length of a route is defined as the number of **through** actions performed by the agent; the final **goto** action is not counted as it does not involve any reasoning.

The agent's task is to find a way from a pre-specified origin s (source location) to a pre-specified destination t (target location). See page 60 for a definition of reaching a destination. It is important to remember that the agent has *only* structural information at its disposal; neither signs nor landmarks are available, even preview evaluation has been turned off because it is not a purely structural feature.

Table 5.2 Wayfinding tasks to be simulated

Task	Env.	Origin	Dest.	Description
1	Enge	**t567**	**p2**	Tramway stop on square to platform 2
2		**p2**	**t567**	reverse direction
3		**t13**	**p2**	Tramway stop on the side to platform 2
4		**p2**	**t13**	reverse direction
5		**sa**	**cr**	Shopping area to circular room
6		**cr**	**sa**	reverse direction
7	Wiedikon	**trs**	**p1**	Tramway stop on square to platform 1
8		**p1**	**trs**	reverse direction
9		**sqs**	**p1**	Lateral square to platform 1
10		**p1**	**sqs**	reverse direction
11		**cafe**	**kiosk**	Café to Kiosk (within building)
12		**kiosk**	**cafe**	reverse direction

There is no point in simulating all origin/destination pairs; the number of such pairs grows quadratically with the number of elements in a Schematic Geometry. Rather, the agent shall find its way between a few well-chosen locations, especially between places of arrival/departure of means of public transport, that is, between tramway stops and platforms. Table 5.2 lists the 12 wayfinding tasks to be simulated, Task 1 through Task 12.

The simulation is performed using Odeon, the implementation of Schematic Geometry and the wayfinding agent (Appendix C). Figure 5.5

shows screenshots of simulated routes. Each task will be performed 1000 times to account for the stochastical moment in the simulation. The output of the simulation is a protocol of what the agent did, including the route travelled and the decisions made along the way. Figure 5.6 shows and explains an example protocol.

Parametrisation. The standard wayfinder allows for a great deal of parametrisation. For the simulation runs from Table 5.2, some parameters were kept constant, while others were varied:

- Previews: turned off (not used by the agent).
- Salience: turned off (all links have same probability being chosen).
- Previous knowledge: **st**, **sq**, **bg**, **bgi**, **pa** (Figure 5.7).
- All combinations of *stayInRoom* and *focusSearch*, except both off, which would turn the standard wayfinder into a random wayfinder.

The random wayfinder and the perfect wayfinder agent have no parameters.

5.6 Results and discussion

The results for the "Enge" tasks are presented in Table 5.3 and those for the "Wiedikon" tasks in Table 5.4. Each row in these tables begins with a task identifier of the form $x.y$ where x references a task from Table 5.2 and y identifies a particular agent type (standard or random) and configuration (SR=*stayInRoom*, FS=*focusSearch*), as given in the second column. Next are statistics over the route length (number of **through** actions performed by the agent); $\bar{U}$ is how often, on average, a unique "choice" remained after applying the heuristics and $\bar{R}$ is how often, on average, the agent had to make a random choice because the heuristics did not reduce the choice set to a unique choice; finally, $\bar{h}_{SR}$ how often, on average, the *stayInRoom* heuristic could be used to reduce the size of the choice set and $\bar{h}_{FS}$ is the same measure for the *stayInRoom* heuristic. Note that $\bar{U} + \bar{R} =$ average route length.

The Enge results (Table 5.3). Immediate observations:

1. Success: in 4 out of 6 tasks, the standard wayfinder with both heuristics (SR and FS) turned on always finds a shortest route (Tasks $1, 3, 5, 6$).
2. Asymmetry: going from **p2** to **t13** is harder than the reverse direction from **t13** to **p2**.
3. The *stayInRoom* heuristic seems more promising than the *focusSearch* heuristic (Task 2 is the only exception).

Is the standard wayfinder *significantly* better than a random wayfinder? For those tasks $(1, 3, 5, 6)$ where the standard wayfinder always finds a shortest route, this is certainly true. For Task 2, a statistical test will help comparing

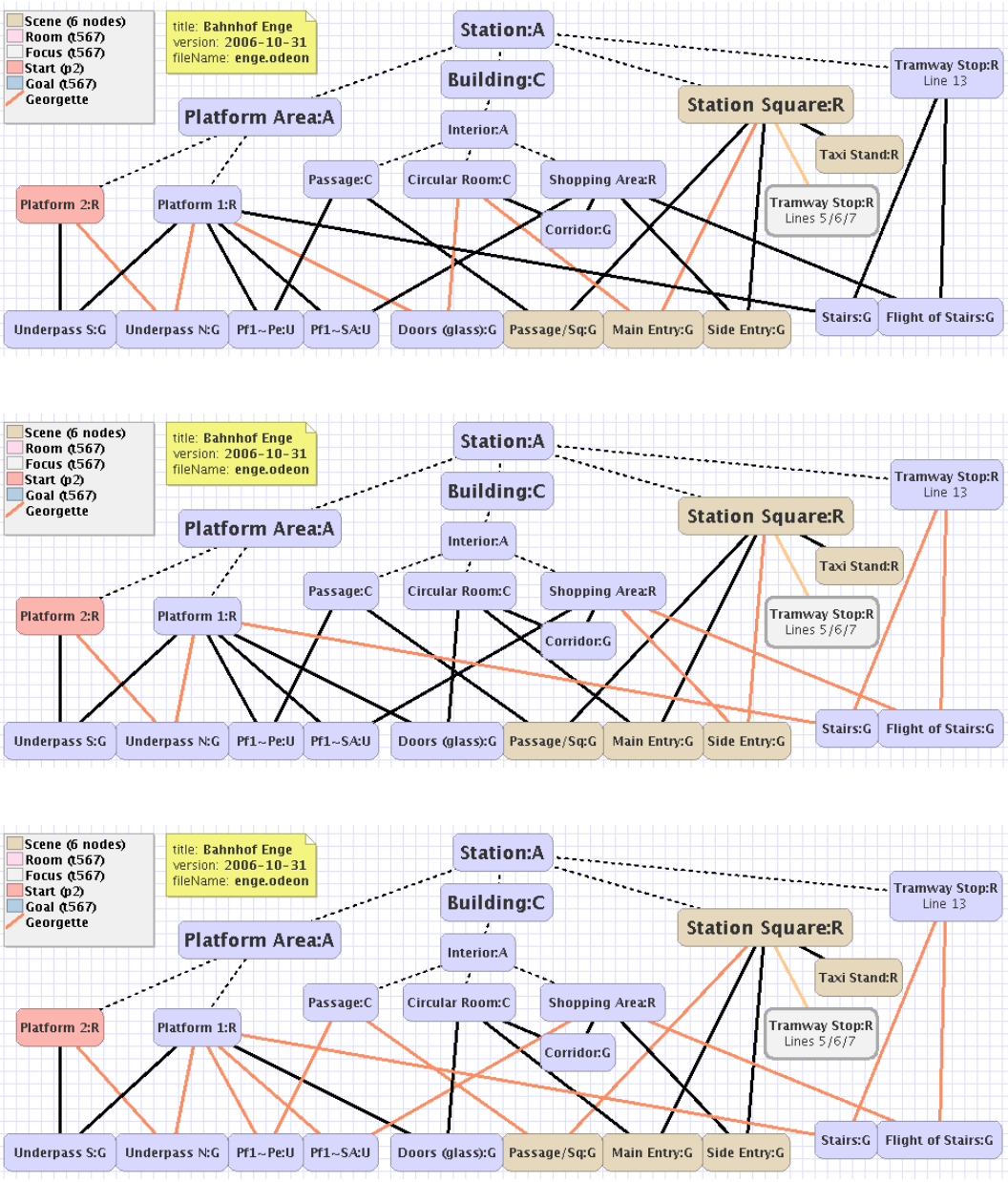


Figure 5.5 Screenshots of Odeon simulating Task 2 (Platform 2 to Tramway Stop at Enge): perfect wayfinder (top), standard wayfinder (middle), and random wayfinder (bottom); the route travelled is indicated with bright red lines. Each task is performed $N = 1000$ times. Results are shown in Tables 5.3 and 5.4.

```
StandardWayfinder Zarif                 Agent type and an arbitrary name
world Bahnhof Enge                      Where the agent lives (see Figure 5.3)
start cr (Circular Room)                  starting place
goal sa (Shopping Area)                   destination to reach
option stayInRoom                       Use the stayInRoom heuristic
option focusSearch                      Use the focusSearch heuristic
option dontGoBack                       Remove last link from choice set
known bgi bg pa sq st                   Previous knowledge (cf. Figure 5.3)

at cr (Circular Room)                   At the agent's current position (cr):
  room bgi (Interior) {known}             the room is known
  where sa (Shopping Area) {unkn.}        but not the room-referent
  link do (Doors to Pf 1) {bad}         Next is a list of all links in
  link co (Corridor) {good}               the current scene along with a
  link me (Main Entry) {bad}              goodLink or badLink indication
  choice 3 stayInRoom 1 unique          Choice set reduced from 3 to 1 (unique)
through co (Corridor)                     using the stayInRoom heuristics and
at sa (Shopping Area)                     the agent goes through co to sa
done L=1 V=13 D=0 U=1 R=0 S=1 F=0 cr,co,sa    Summary line
```

The agent reached the goal by crossing $L = 1$ link, making $U = 1$ unique choice and $R = 0$ random choices, encountering $D = 0$ dead ends, applying once the *stayInRoom* heuristics ($S = 1$) and never the *focusSearch* heuristics ($F = 0$). The full route was **cr**→**co**→**sa** and the agent's memory now contains $V = 13$ elements.

Figure 5.6 Agent protocol

the route lengths. We have two independent and unpaired samples that do not follow a normal distribution. This is a typical application for Wilcoxon's rank sum test for H_0: $\Pr(X_1 > X_2) \geq \frac{1}{2}$ against H_1: $\Pr(X_1 > X_2) < \frac{1}{2}$. This test is sensitive to differences in the median but not in the variance. Here X_1 is the standard wayfinder's route length and X_2 the random wayfinder's route length. We get these figures:

	Min	Q1	Median	Mean	Q3	Max
Std:	3	3	3	3.75	4	10
Rnd:	3	5	8	10.05	13	85

$$W = 164633.5 \quad p < 0.001$$

W is the test statistic and p is the so called "p-value," the probability that we get W by chance if H_0 is true. Since p here is very low, we can savely

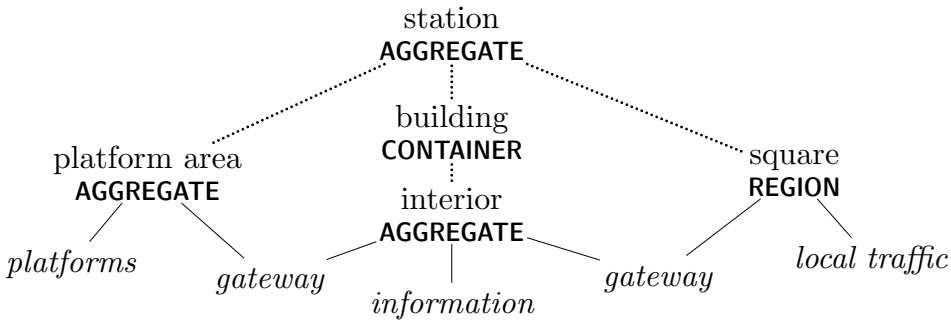


Figure 5.7 Assumed general knowledge as a subset of the complete schematic structure of a railway station. The known elements are named **st**, **pa**, **bg**, **bgi**, and **sq** in the Enge and Wiedikon datasets. Elements in *italic type* indicate further knowledge that is not a subset of the complete Schematic Geometry model for the station. It is not used in the simulations, but hints at future work. Especially the presence of *information* in the building is a source for another wayfinding heuristic: if in doubt, go into the station building; chances are, there is information about how to reach the goal.

reject H_0 and state that the standard wayfinder is significantly better than the random wayfinder in Task 2.

The problem with Task 4 is that $\bar{h}_{SR} = 0$ and $\bar{h}_{FS} = 0$: the standard wayfinder's two main heuristics could never be *effectively* applied.[17] More precisely, the location of the tramway stop **t13** on the side of the station (instead of on the station square) is a non-standard situation and not compatible with the agent's previous knowledge (Figure 5.7). If we add **t13** to the agent's knowledge, then it reliably finds a way from **p2** to **t13** (Task 4.5 in Table 5.3).

A particularly interesting finding is the asymmetry in the agent's performance: going from A to B is not the same as going from B to A, as is evident when comparing the standard wayfinder's performance in Tasks 1 and 2, or in Tasks 3 and 4 (Table 5.3). The reason is again in the table's last two columns: in Tasks 2 and 4, $\bar{h}_{SR} = 0$, that is, *stayInRoom* could never be effectively used for wayfinding; in Task 4, $\bar{h}_{FS}$ is also zero. This relates to the structure of the station, which will be analysed in Chapter 6.

Which of the two heuristics is more useful, *focusSearch* (the $x.2$ tasks) or *stayInRoom* (the $x.3$ tasks)? A quick look at Table 5.3 indicates that

[17] That is, the heuristics did not reduce the choice set or they could not be applied at all because Rumelhart's room (for *stayInRoom*) or room-referent (for *focusSearch*) is unknown.

Table 5.3 Results of the Enge simulation runs ($N = 1000$)

Task 1: **t567→p2** ($L_p = 3$)

Task	Agent	Min	Q1	Med	Q3	Max	Mean	$\bar{U}$	$\bar{R}$	$\bar{h}_{SR}$	$\bar{h}_{FS}$
1.1	Std SR FS	3	3	3	3	3	3	1	2	1	1
1.2	Std – FS	3	3	5	7	31	6.18	2.59	3.59	–	2.59
1.3	Std SR –	3	3	3	4	10	3.78	0.58	3.20	1	–
1.4	Random	3	6	11	19	85	14.54	–	–	–	–

Task 2: **p2→t567** ($L_p = 3$)

2.1	Std SR FS	3	3	3	4	10	3.75	1.52	2.32	0	1
2.2	Std – FS	3	3	3	4	10	3.78	1.51	2.66	–	1
2.3	Std SR –	3	3	5	7	28	6.00	1.43	4.57	0	–
2.4	Random	3	5	8	13	85	10.05	–	–	–	–

Task 3: **t13→p2** ($L_p = 2$)

3.1	Std SR FS	2	2	2	2	2	2	1	1	1	1
3.2	Std – FS	2	2	4	6	28	5.16	2.58	2.58	–	2.58
3.3	Std SR –	2	2	$2\frac{1}{2}$	4	12	3.28	0.17	3.11	1	–
3.4	Random	2	4	9	17	64	12.55	–	–	–	–

Task 4: **p2→t13** ($L_p = 2$)

4.1	Std SR FS	2	4	7	13	61	9.57	1.75	7.82	0	0
4.2	Std – FS	2	3	7	12	54	9.36	1.64	7.72	–	0
4.3	Std SR –	2	3	6	12	65	9.24	1.61	7.63	0	–
4.4	Random	2	4	8	15	73	10.99	–	–	–	–
4.5	Std*SR FS	2	2	2	2	2	2	1	1	0	1

Task 5: **sa→cr** ($L_p = 1$)

5.1	Std SR FS	1	1	1	1	1	1	1	0	1	0
5.2	Std – FS	1	1	2	4	22	3.29	0.68	2.62	–	0.85
5.3	Std SR –	1	1	1	1	1	1	1	0	1	–
5.4	Random	1	1	4	9	42	6.16	–	–	–	–

Task 6: **cr→sa** ($L_p = 1$)

6.1	Std SR FS	1	1	1	1	1	1	1	0	1	0
6.2	Std – FS	1	1	2	4	16	2.86	0.42	2.44	–	0.61
6.3	Std SR –	1	1	1	1	1	1	1	0	1	–
6.4	Random	1	1	3	6	32	4.42	–	–	–	–

L_p=perfect wayfinder's (=minimal) route length
Med=median, Q1=first quantile, Q3=third quantile
See text for $\bar{U}$, $\bar{R}$, $\bar{h}_{SR}$, and $\bar{h}_{FS}$; SR=*stayInRoom*, FS=*focusSearch*
* Task 4.5 is with **t13** initially known

stayInRoom is always better, except in Task 2, where *focusSearch* is better. Statistical analysis confirms this observation (Wilcoxon test, H_0: equal median $\tilde{x} = \tilde{y}$ and H_1: $\tilde{x} > \tilde{y}$):

	FS	SR	Wilcoxon test results		
Task 1:	$\tilde{x} = 5$	$\tilde{y} = 3$	$W = 688848$	$p < 0.001$	
Task 2:	$\tilde{x} = 3$	$\tilde{y} = 5$	$W = 316388$	$p < 0.001$	(H_1: $\tilde{x} < \tilde{y}$)
Task 3:	$\tilde{x} = 4$	$\tilde{y} = 2.5$	$W = 629933.5$	$p < 0.001$	
Task 4:	$\tilde{x} = 7$	$\tilde{y} = 6$	$W = 507551$	$p = 0.278$	
Task 5:	$\tilde{x} = 2$	$\tilde{y} = 1$	$W = 867000$	$p < 0.001$	
Task 6:	$\tilde{x} = 2$	$\tilde{y} = 1$	$W = 829000$	$p < 0.001$	

The problem with Task 4 (**p2→t13**) is that *stayInRoom* cannot reduce the choice set because Rumelhart's room is always the greatest element (the "station") and thus everything is within the room. Neither does *focusSearch* help: it can never be applied because the room-referent (**t13**) is not known and cannot be known until it is reached by the agent. Therefore, the agents of Tasks 4.1, 4.2, and 4.3 are essentially random wayfinders.

The Wiedikon results (Table 5.4). The first thing to note is that none of the agents in none of the tasks always finds a shortest route. Moreover, the *stayInRoom* heuristic can never be effectively applied in Tasks 7 through 10. Is the standard wayfinder (with both heuristics turned on) any better than a random wayfinder? A glance at the average route length and the median supports this and Wilcoxon's rank sum test shows it is even highly significant (H_0: equal median $\tilde{x} = \tilde{y}$ and H_1: $\tilde{x} > \tilde{y}$):

	Std	Rnd	Wilcoxon test results	
Task 7:	$\tilde{x} = 5$	$\tilde{y} = 13$	$W = 176546$	$p < 0.001$
Task 8:	$\tilde{x} = 5$	$\tilde{y} = 11$	$W = 211973.5$	$p < 0.001$
Task 9:	$\tilde{x} = 4$	$\tilde{y} = 10$	$W = 248193$	$p < 0.001$
Task 10:	$\tilde{x} = 10$	$\tilde{y} = 14$	$W = 389057.5$	$p < 0.001$
Task 11:	$\tilde{x} = 4$	$\tilde{y} = 8$	$W = 338687.5$	$p < 0.001$
Task 12:	$\tilde{x} = 4$	$\tilde{y} = 8$	$W = 354069$	$p < 0.001$

Even if the standard wayfinder is better than a random wayfinder, it is not particularly good: its average route length is, in all tasks, at least twice as long as the shortest route and the median (which is less sensitive to outliers) is only in Tasks 7 and 8 less than twice the shortest route's length.

Table 5.4 Results of the Wiedikon simulation runs ($N = 1000$)

Task 7: **trs→p1** ($L_p = 3$)

Task	Agent	Min	Q1	Med	Q3	Max	Mean	$\bar{U}$	$\bar{R}$	$\bar{h}_{SR}$	$\bar{h}_{FS}$
7.1	Std SR FS	3	4	5	8	26	6.21	2.05	4.17	0	1.23
7.2	Std – FS	3	4	5	8	21	6.21	2.03	4.18	–	1.27
7.3	Std SR –	3	5	8	13	56	9.84	3.03	6.81	0	–
7.4	Random	3	8	13	23	134	17.79	–	–	–	–

Task 8: **p1→trs** ($L_p = 3$)

8.1	Std SR FS	3	3	5	7	27	5.99	1.72	4.27	0	1
8.2	Std – FS	3	3	5	7	27	5.64	1.67	3.97	–	1
8.3	Std SR –	3	4	6	9	49	7.48	1.62	5.86	0	–
8.4	Random	3	6	11	19	82	14.65	–	–	–	–

Task 9: **sqs→p1** ($L_p = 2$)

9.1	Std SR FS	2	2	4	7	29	5.43	1.34	4.09	0	1.27
9.2	Std – FS	2	4	4	7	23	5.58	1.40	4.18	–	1.25
9.3	Std SR –	2	4	7	12	78	8.91	2.25	6.66	0	–
9.4	Random	2	5	10	20	132	15.35	–	–	–	–

Task 10: **p1→sqs** ($L_p = 2$)

10.1	Std SR FS	2	5	10	18	64	12.57	2.45	10.12	0	0
10.2	Std – FS	2	5	10	17	90	13.01	2.59	10.41	–	0
10.3	Std SR –	2	5	10	17	96	13.18	2.61	10.57	0	–
10.4	Random	2	7	14	25	143	19.16	–	–	–	–

Task 11: **cafe→kiosk** ($L_p = 2$)

11.1	Std SR FS	2	2	4	8	36	5.89	2.95	2.95	5.89	0
11.2	Std – FS	2	2	3	13	106	9.02	2.37	6.65	–	0
11.3	Std SR –	2	2	4	8	46	5.70	2.85	2.85	5.70	–
11.4	Random	2	3	8	19	157	15.12	–	–	–	–

Task 12: **kiosk→cafe** ($L_p = 2$)

12.1	Std SR FS	2	2	4	8	40	6.05	3.02	3.02	6.05	0
12.2	Std – FS	2	2	5	13	59	8.66	2.22	6.44	–	2.87
12.3	Std SR –	2	2	4	8	36	6.10	3.05	3.05	6.10	–
12.4	Random	2	3	8	21	180	15.02	–	–	–	–

L_p=perfect wayfinder's (=minimal) route length
Med=median, Q1=first quantile, Q3=third quantile
See text for $\bar{U}$, $\bar{R}$, $\bar{h}_{SR}$, and $\bar{h}_{FS}$; SR=*stayInRoom*, FS=*focusSearch*

The asymmetry that was found in the Enge scenario (wayfinding from A to B is not the same as from B to A) is only present in Tasks 9 and 10 (**sqs**↔**p1**), but it is statistically significant (Wilcoxon's rank sum test):

	Min	Q1	Median	Mean	Q3	Max
Task 9:	2	27	4	5.43	7	29
Task 10:	2	5	10	12.57	18	64

$$W = 256245.5 \quad p < 0.001$$

Tasks 11 and 12 are the only tasks examined where the agent can effectively apply the *stayInRoom* heuristic. The room is the building's interior, which is initially known by the agent (see Figures 5.4 and 5.7). This heuristic keeps the agent inside the building (**bgi**), which is good, but once in he **hall**, there are still several options and *focusSearch* cannot be applied because the destination's room-referent (the café or the kiosk) is not known.

When evaluating route lengths, it is also worth comparing them with the longest shortest path in the emergent network, measured as the number of required **through** actions (so it is comparable to the agent's reported route length). This is 4 for the Wiedikon station (namely, from **sq** to **br**) and 3 for the Enge station (from **sq** to **p2**) and means that the standard wayfinder does considerable walks compared to the size of its environment! Nevertheless, the standard wayfinder is in both scenarios significantly better than a random wayfinder.

Of course, a real human wayfinder would use many other sources of information besides the schematic structure of a building and therefore is likely to perform much better than the standard wayfinder, even on first encounters with the Wiedikon station. Within the context of this thesis, however, only the structure is of interest and shall be analysed in the next chapter.

Chapter 6

Discussion

Results from the simulation runs in the previous chapter show:

1. The standard wayfinder is significantly better than a random wayfinder, but often considerably worse than a perfect wayfinder.
2. Going from A to B can be considerably harder than going from B to A.
3. At Enge, *stayInRoom* is more useful than *focusSearch*, whereas at Wiedikon, *stayInRoom* often could not be used at all.

The first result shows that structural information alone certainly helps with wayfinding but does not completely solve the problem. Humans are integrating and using information from many different sources and are therefore expected to perform much better than the software agent using only structural information.

The second result is particularly noteworthy: asymmetry was not explicitly coded into the model but nicely corresponds with experimental findings [39,77:83] and theory [128]. What precisely is the reason for this asymmetry in the agent's performance?

There are also counter-intuitive peculiarities: Why scored the agent so badly at the seemingly simple Task 11 (Café to Kiosk at Wiedikon) while the seemingly complex Task 1 (Tramway Stop to Platform 2 at Enge) is always solved perfectly? And why can *stayInRoom* only rarely be used at Wiedikon?

The remainder of this chapter evaluates the experimental results and the methodology employed. Section 6.1 substantiates and explains the observations stated above. Section 6.2 considers other wayfinding strategies than those implemented in the standard wayfinder. Section 6.3 compares wayfinding in scene space to wayfinding in network space. Finally, Section 6.4 evaluates the adequacy of Schematic Geometry for representing scene space.

6.1 Wayfinding in scene space

To evaluate my approach to modelling wayfinding in scene space, I follow the information sources available to real humans when wayfinding:

spatial structure R
salience/landmarks R
previous knowledge D&R
signage/previews R
timetable D

where D means useful for destination choice and R means useful for route choice (en-route decision-making). While the timetable is specific to wayfinding in public transport, the other information sources equally apply to other environments and shall now be discussed, followed by a consideration of other wayfinding strategies and a summary of structural information use.

Spatial structure originates from the architectural layout: linking and nesting of cognitive spatial schema instances and sometimes also previews through gateways to elements beyond. Structural information is the standard wayfinder's only source of information for decision-making. The use of previews, however, was turned off in the simulations because previews are not purely structural—they also depend on transparent materials like glass doors.

When the standard wayfinder performed well, then it managed to use structural information to its advantage. For example, going from the tramway stop to platform 2 at Enge is always solved perfectly by the standard agent, even though it is a comparatively long route ($L = 3$, which equals the longest shortest path at Enge). The agent always takes one of these routes:

```
t567,sq,me,cr,do,p1,un,p2
t567,sq,me,cr,do,p1,us,p2
t567,sq,pq,pe,pp,p1,un,p2
t567,sq,pq,pe,pp,p1,us,p2
t567,sq,se,sa,ul,p1,un,p2
t567,sq,se,sa,ul,p1,us,p2
```

Compare with Figure 5.3; these routes were extracted from the protocols generated during the simulation (Figure 5.6). Each route contains three links (therefore $L = 3$) and corresponds to a shortest route (not metrically shortest, of course, but in terms of the links to be taken).

When at **t567**, looking around takes the agent "up" to the whole station square **sq**, from where there are three gateways, all leading into the

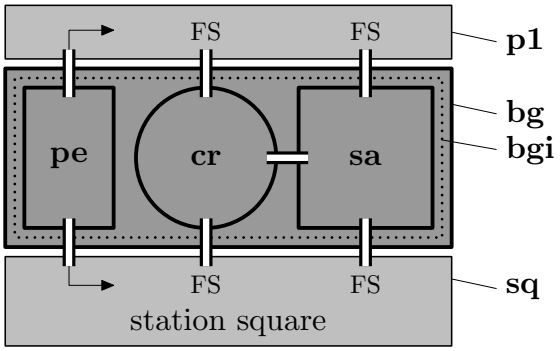


Figure 6.1 When going from **pe** to **sa**, *stayInRoom* must be violated because there is no in-house connection. Once on the platform **p1** (or the station squrare **sq**), *focusSearch* tells the agent to go back into the building.

building. Using only structural information, they are all equivalent, so the agent chooses one at random:

1. through **pq** into the passage **pe**, from where there is only one gateway other than the one the agent came in, so the choice is unique: through **pp** to **p1**.
2. through **me** into the circular room **cr**, from where there are two further gateways: **do** out to platform 1 and **co** to the shopping area within the building. The room is the station, so *stayInRoom* does not help, but the room-referent is the platform area, which is known by the agent (Figure 5.7), so *focusSearch* tells the agent to go through **do**.
3. through **se** into the shopping area **sa**, where *focusSearch* can be applied for the same reason as in the previous case.

Once on Platform 1 (**p1**), the room is the platform area (**pa**) and the only links that keep the agent within the room are the two underpasses; either takes the agent to the goal. The agent was so successful in this task, because at any step, at least one of his heuristics could be successfully applied or all options were equivalent (the first step). Using previews could not have improved this result.

What about going from the Passage **pe** to the shopping area **sa** at Enge? This is a contrived example and therefore was not simulated in Chapter 5, but shall now be investigated (see Figure 6.1). When in **pe**, the room is the interior of the building and the room-referent is not known; *focusSearch* cannot be applied and *stayInRoom* has to be violated because there are no other options! Randomly, the agent goes out to Platform 1 or to the station square. Leaving the current room means that the next room is larger, and

the same holds for the room-referent: in both cases, the room is now the whole station and the room-referent the whole building, which means that the agent is conceptually farther from the goal and now tries to get back into the building (the *focusSearch* heuristic). In both cases (platform and shopping area) there are two options for doing so (a third option would be to go back where the agent came from, but that is ruled out by the *dontGoBack* heuristic). Both options are structurally the same, so the agent chooses one at random. A simulation confirms that the standard wayfinder actually takes the four routes just discussed with about equal probabilities:

```
pe,pp,p1,do,cr,co,sa
pe,pp,p1,ul,sa
pe,pq,sq,me,cr,co,sa
pe,pq,sq,se,sa
```

When the standard wayfinder performs badly, there is not enough information available, or it did not exploit it properly. For example, going from the Café to the Kiosk at Wiedikon (Task 11 in the previous chapter; see Table 5.4 and Figure 5.4) turned out to be difficult: the standard wayfinder's average route length was more than twice the minimum route length and longer than the longest shortest path in that station. An examination of the agent protocols reveals that all simulation runs start with the sequence **cafe**-**h2c**-**hall**. The *stayInRoom* heuristic forbids going through **ce** out of the building. Once in the hall, the agent has three options: **h2k**, **ss**, and **aws**; the main entry **me** is again ruled out by *stayInRoom*. Using only structural information and the knowledge shown in Figure 5.7, there is no way of choosing one of the gateways other than at random: *focusSearch* cannot be applied because the room-referent (**kiosk**) is not known and with respect to *stayInRoom* all gateways are acceptable (Figure 6.2).

What would have helped the agent in this situation? First, a sign identifying **h2k** as the door into the kiosk, but that is signage and not structural information. Second, the same effect can be achieved with a preview from the hall through the door into the kiosk. In reality, both clues exist, a sign and a preview (the doors are transparent). Moreover, all sorts of newspaper racks in front of the door are another indication that this particular door leads into the kiosk; this is, however, not structural but ontological information! Finally, knowledge about the destination's room-referent while in the hall would have allowed use of the *focusSearch* heuristic. As can be verified using the simulation software, the agent then always finds the perfect route

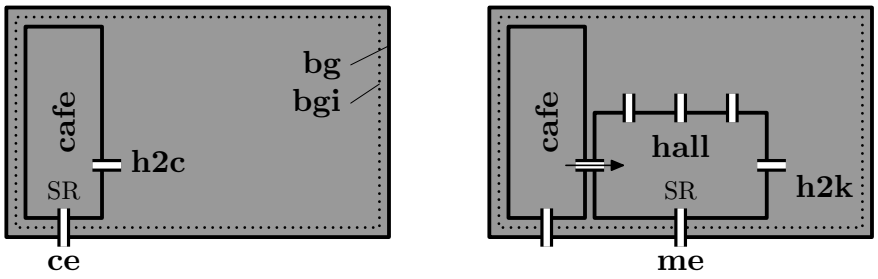


Figure 6.2 Going from **cafe** to **kiosk** (Task 11); we have room=**bgi** (known) and room-referent=**kiosk** (unknown). The first step is clear because *stayInRoom* rules out **ce**. But once in the hall, there are four structurally equivalent options (**h2c** is ruled out by *dontGoBack* and **me** is ruled out by *stayInRoom*).

cafe-h2c-hall-h2k-kiosk of length $L = 2$ with one application of *stayInRoom* (to go through **h2c** into the hall) and one application of *focusSearch* (to go through **h2k** into the kiosk).

Why, then was the agent so much better in going from **sqs** to **p1** at Wiedikon than in the reverse direction? (Table 5.4) Going from **sqs** to **p1** requires entering the building as there is no direct access to the platforms. Once in the building at **corr** (where the elevators go to the platforms) or on **lg** (where the stairs go to the platforms), a *focusSearch* will help because the room-referent (the platform area) is part of the agent's previous knowledge. In the reverse direction, this heuristic will not help because the Side Square **sqs** is not known by the agent until it is reached. The other example where the agent's performance differs significantly between two directions is **p2↔t13** at Enge (Tasks 3 and 4). The reason is again that, due to lacking prior knowledge, *focusSearch* cannot be applied. This suggests that asymmetry in the agent's performance is mainly due to prior knowledge and not to perceived spatial structure. This also means that signage (or previews) are essential for "non-standard" destinations, that is, those that do not fit into the general structure of a station, as the one assumed in Figure 5.7.

Salience and landmarks. Landmarks help identify places in space, but they need prior knowledge to be re-cognised. Landmarks were found to be important in knowledge about urban space [77]. Landmarks help humans to orient themselves, they can serve as an anchor for all sorts of associations, including affections, and they trigger actions in a travel plan. Landmarks are rarely found in railway stations and therefore not considered in this work.

Landmarkedness is related to salience, the quality of standing out against a background. More generally, salience can also mean standing out

against several elements of the same kind. For example, a building's main entry usually stands out against other entrances by way of its architectural design or ornamentation. Determining an object's salience is subject to ongoing research [12]. Salience is not used in the present work, but the simulation tool allows the assignment of salience values to gateways. For example, in the Wiedikon scenario (Section 5.4) the main entrance could be given a higher salience value than the other entrances. If the standard wayfinder is configured to use salience, then it will go through the main entrance with a higher probability than through the other entrances.

Use of previous knowledge. Rather than (unrealistically) assuming that travellers have no knowledge at all, the standard wayfinder is equipped with some general knowledge about the structure of a railway station (Figure 5.7). As is evident from comparing Task 4.1 and Task 4.5 (see Table 5.3), the amount of previous knowledge can have a great impact on the agent's wayfinding performance. It is now necessary to review how previous knowledge enters the agent's wayfinding process and influences its performance.

I assume that if an element is known (by the agent or a human), then it will be recognised (even though it probably cannot be recalled). I further assume that this recognition also triggers recall of the element's location (in terms of the "belongs-to" relation, see Section 4.2), provided that this location is also known. This assumption is motivated by the fact that hierarchy is a common mnemonic device [84]. Not all nesting levels have to be known: seeing a kiosk we may only recall that it belongs to the building, but not where precisely it is located. Schematic Geometry readily supports this type of incomplete knowledge (see the standard wayfinder pseudo code on page 65).

The implication is that the two Room Theory heuristics can only be applied if the room and/or the room-referent are known, otherwise the agent would not be able to recognise them. Tasks 4.1 and 4.5 in the previous chapter illustrate this impact. Humans are probably better at recognition; for example, by relating something to common sense knowledge in compensation for lacking previous knowledge.

A useful property of the Room Theory heuristics is that they build on most general knowledge about the environment. The room is the least upper bound of the agent's current place and the destination, and the room-referent is one step below. This maximises their applicability given only minimal previous knowledge. But it also means that parts of a station that do not fit into the agent's prior knowledge will only be found by chance (or by using other information sources such as previews or signage).

This means that designs of architectural spaces should stick to a common layout whenever possible. Structural similarity supports wayfinding by maximising the traveller's benefit from previous knowledge.

Signage and previews. Signage is an environmental source of wayfinding information, but the information provided is not structural. Instead, signage identifies objects (for example, identifying a particular platform as "Platform 1") and directs the traveller along a particular way (in Figure 5.4, for example, the words "to the trains" (in German) appear above the stairs that lead down to the platforms of the Wiedikon station).

The latter example illustrates that pointing signs are in essence identifiers too: they identify a particular gateway or path or direction. Signs that merely identify a direction are often prone to ambiguity. More useful (and robust) are pointing signs that directly identify a path or a gateway along the route to the intended destination. Empirical research that supports this claim is lacking to date.

Signs must be understood to be useful. Symbols in place of text overcome language barriers, and standard symbol sets have been developed.[18] But even with well designed symbols, the danger of an information overload [4:34] is lurking: wayfinding-related signs are often in competition with a multitude of marketing-related signs.

As humans are excellent at object recognition [17], *the best sign is a preview* to the destination. Simulations with the standard wayfinder and the previews function turned on illustrate the efficiency of previews for wayfinding. For example, at Wiedikon the side entry **se** is of glass and offers a preview of the side square **sqs** from the corridor **corr**. The result of $N = 1000$ simulation runs with the same settings as in Task 10 (**p1**→**sqs**, $L_p = 2$) of Chapter 5 but the preview added gives a dramatic improvement, as these figures show:

	Min	Q1	Median	Q3	Max	Mean
Chapter 5:	2	5	10	18	64	12.57
with preview:	2	2	4	6	27	5.05

As far as the standard wayfinder is concerned, pointing signs can be treated as previews. It is therefore easy to simulate the effect that the abovementioned "to the trains" sign has on wayfinding performance at Wiedikon

[18] See, for example, the set of 50 transportation-related symbols, developed in the 1970ies by the U.S. Department of Transportation and the American Institute of Graphic Arts, freely available from `http://www.aiga.org/`

station. This sign is essentially a pointer from **hall** through **ss** to **lg**. The results of $N = 1000$ simulation runs with the same settings as in Task 7 (**trs**→**p1**, $L_p = 3$) but the preview as discussed are:

	Min	Q1	Median	Q3	Max	Mean
Chapter 5:	3	4	5	8	26	6.21
with sign:	3	3	4	6	6	4.50

Previews are the abstract schematic (non-geometrical) counterpart of lines of sight. Lines of sight are an important foundation for the Space Syntax toolkit [47]. But the notion of a line of sight is a purely geometrical concept (compare Galton's line-of-sight calculus [33]) and is not meaningful per se. In contrast, previews are about what can be seen (and recognised) along a line of sight by pointing from one place in a Schematic Geometry model through a gateway to another place.

Previews are often through windows, not through gateways. Such previews help with knowledge acquisition but not with immediate wayfinding decisions because windows cannot be walked through. For example, the line of sight in the Wiedikon station (Section 5.4) gives the traveller entering the building through the side entry a preview to the Café. The traveller now knows about the existence of the Café and has a general idea about the direction to the Café. But without further assumptions (like a least-angle heuristic [50]) it is still not clear where to go to reach the Café.

Summary of structural information use. The scene graph defines the options for the wayfinder's next "step" (qualitative change of location): in every scene there are connections to other scenes (targets for **through** actions) and possibly objects and regions within the scene (targets for **goto** actions). The hierarchical structure is used for decision-making as follows:

- Inheritance of location (dominance, page 51) is used for going up in the hierarchy of nested elements.
- Scenes prevent the wayfinder from going up in the part-of hierarchy of spatial elements beyond that what is *practically* possible: when you are in the circular room of the station in Figure 5.3, then you are also in the building and in the station aggregate, but both are outside of your current scene. Nevertheless, they are available for reasoning (given that they are known) using the two Room Theory heuristics.
- The *stayInRoom* heuristic keeps the wayfinder in a particular branch of the hierarchy (more precisely: the room's lower bound in the poset).

– The *focusSearch* heuristic is for changing to another branch in the hierarchy (namely, into the room-referent or its lower bound).
– The destination is reached once the current position equals the destination or properly belongs to the destination (page 60).

6.2 Other wayfinding strategies

The two Room Theory heuristics fully exploit the hierarchical structure of scene space. Other wayfinding heuristics are conceivable, but they are likely not purely structural.

For example, going into the "main" station building is a reasonable heuristic because there is usually a great deal of information and the building is usually highly integrated with other places. But what is the main station building? This is more an ontological than a structural question and therefore hard to answer within Schematic Geometry. Humans probably have a general idea of the "look" of a main station building and usually there is a sign (of type 1) that reads "station." A purely structural approach could be to look for the largest **CONTAINER** (which works fine for the two examples of Chapter 5 but fails if there is more than one).

Another heuristic is to maintain a direction, as Raubal's agent [100] does. Whenever a distant destination (or intermediate destination) can be seen (such as the Café at Wiedikon by means of the line-of-sight, p. 70), the wayfinder gets an idea of the direction and can use path integration (p. 17). However, path integration is a geometrical concept, not a structural one and thus not easy to implement with Schematic Geometry Raubal uses a graph whose nodes are embedded into the Euclidean plane and thus can easily use direction.

Finally, the standard wayfinder can learn more while moving around, thus enhancing the chance that a preview or a *focusSearch* can be applied. For example, when standing on the station square, the current scene includes the entrance into the station building (it is visible and reachable) but excludes the station building (it is not reachable other than by going through the entrance). In general, all **CONTAINER** instances bounding a given **REGION** instance are visible to the wayfinding agent and therefore available to his/her reasoning. The precise meaning of a "bounding" **CONTAINER** awaits formal definition within Schematic Geometry.

6.3 Comparison with network space

Wayfinding in scene space is different from wayfinding in network space. In scene space, global information is locally available because of the hierarchical structure; the two Room Theory heuristics for wayfinding exploited this kind of structural "knowledge in the world." In contrast, networks do not intrinsically provide information about the global structure; such information must come from other sources, typically signage for en-route wayfinding and network maps for pre-trip planning.[19]

Section 4.3 showed that there is a network—though not an obvious one—even in scene space. It is called the scene graph and emerges from an analysis of the schematic structure of scene space. This network is cognitively motivated and not simply imposed. But as in network space, movement in scene space takes place along the emergent network. This allows traditional network methods to be applied to scene space and wayfinding can be understood as routing in a network, as usual. For example, the perfect wayfinder used a shortest path algorithm to find the perfect route. Keep in mind that the scene graph consists of *spatially extended nodes*, whereas the *links have no spatial extent* (they merely express parthood); precisely the opposite is true in network space whose structure is a "classical" network.

Robustness. The standard wayfinder has two features that allow for robust wayfinding. First, the two Room Theory heuristics either pull the agent towards the destination (*focusSearch*), keep it in the goal's vicinity (*stayInRoom*), or do not apply at all; never will the agent "overshoot" the destination.[20] This has to be contrasted with the use of a route description, which is tailored to a particular route and is useless or even misleading, should the designated route ever be lost.

Second, the standard wayfinder never plans a complete route! Instead, it follows Wiener's *fine-to-coarse* planning hypothesis [134]. An environment that affords successful navigation without planning complete routes offers a

[19] Despite frequent mention of "hierarchical" networks, networks are essentially flat: at every node there is only information about incident links. The hierarchy in those networks is always attributed, not structural. For example, a particular link in a transport network may be tagged as "motorway" and another link as "street," but never is a "street" link also a "motorway" link. Nevertheless, when *designing* networks it may be useful to think of them hierarchically [92].

[20] Note that the *dontGoBack* heuristic could potentially push the agent away from the destination, but as it is applied only *after* the two Room Theory heuristics (see code on page 65), this negative effect never happens.

great level of comfort and quality to the traveller (because only little cognitive resources have to be devoted to the wayfinding task) and allows for *robust* navigation (because there is no need for a plan that could be destroyed by interruptions of all sorts [68]).

Both, the Room Theory heuristics and Wiener's fine-to-coarse hypothesis, rely on the hierarchical structure of scene space and are therefore not possible in (non-hierarchical) network space. In practice, however, a network is always embedded in an environment that provides auxiliary (probably hierarchical) information. If one were to drive to the city and the scenery is changing from urban to rural along the way, this would be a strong indication to alter the route. This example shows an application of the *stayInRoom* heuristic in network space, but using information that has nothing to do with the network structure. In a similar vein, signage can be seen as the information source for a network space equivalent of the *focusSearch* heuristic, but again, signage has nothing to do with the network structure.

In summary, the hierarchical scene space structure allows for efficient wayfinding with minimal planning and therefore maximal robustness. The same can be achieved in network space, but there one has to resort to information beyond the network structure. Using such information in scene space can further improve wayfinding performance.

6.4 Schematic Geometry reviewed

Schematic Geometry is for scene space what graph-based models are for network space: a representation of the prevalent structure, careful not to impose any new structure. Schematic Geometry is firmly grounded in human cognition by its image schematic foundation. But is the resulting formalism really adequate (*i*) as a model of space, and (*ii*) as a model of knowledge of space? How does it compare to similar abstractions?

According to Mazzola [81:440], "knowledge is ordered access to information," that is, the mental organisation of pieces of information matters. Miller explicitly proposes hierarchy to be such an organisational principle (a "mnemonic device" in his words) [84]. Schematic Geometry as a model of knowledge about (scene) space encodes both, the information (in terms of cognitive spatial schemata) and its organisation (in terms of the hierarchy). The fact that a hierarchy emerges from the structure of the underlying image schemata is a strong point in case that image schemata truly represent knowledge.

Schematic Geometry is essentially a partonomy. The agent simulation assumed it to be transitive, but the transitivity of partonomies is debated

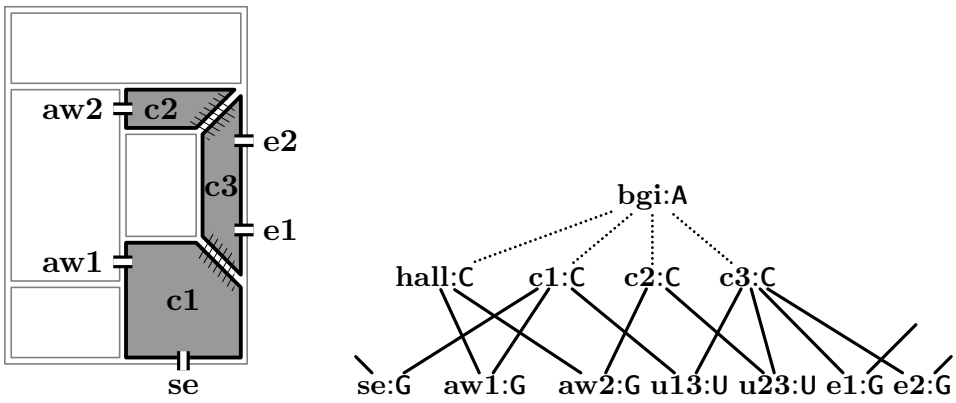


Figure 6.3 Alternative representation of the corridor at Wiedikon—compare with Figure 5.4. This representation is correct but needlessly complicated and results in the standard wayfinder performing worse.

[132]. For example, while the main entry is a functional part of a station building, it is hardly a functional part of the whole station aggregate; nevertheless, it is part of the whole station in a spatial sense. This example suffices to illustrate the problem and its solution: specific types of part-of relations (such as *functional*-part-of) may not be transitive, but the generic part-of relation always is. For this reason, Section 4.2 only generally describes the part-of relation as one of "belonging to" a larger entity. As the experiments illustrate, this generality is not a problem for simulating wayfinding.

There is some latitude in Schematic Geometry modelling. For example, the two station squares at Wiedikon (**sq** and **sqs**) could be seen as just one REGION without the need for a ULINK. At Enge, there is not much room for such debate because its station square is convex. However, the weird shape of the shopping area could be used as an argument for splitting it into several REGIONS, connected to each other by ULINKS. Other elements, especially gateways and containers, are very unlikely to be debated, because they correspond so well with established image schemata. For regions and unconscious links, the correspondence is less immediate.

What is the influence of such potential structural changes to the agent's wayfinding performance? Redoing some simulations from Chapter 5 should answer this question. Assume that the Corridor (**corr**) at Wiedikon is represented as three convex REGIONS, connected by ULINKS (Figure 6.3). Simulating Task 11 (wayfinding from **cafe** to **kiosk**; $N = 1000$ runs) now in this modified environment yields an average route length of 11.46, which is considerably longer than in the original model.

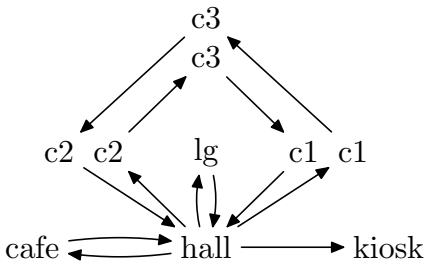


Figure 6.4 Sometimes the expected route length can be directly computed. The figure shows the state graph of the standard wayfinder going from **cafe** to **kiosk** in the modified Wiedikon environment (Figure 6.3) with the *dontGoBack* option turned off. Let L be the travelled route length from **cafe** to **kiosk** and L' the travelled route length from **hall** to **kiosk**. Then we have $\mathrm{E}\,L = 1 + \mathrm{E}\,L'$ and $\mathrm{E}\,L' = \frac{1}{5}\cdot 1 + \frac{2}{5}\cdot(\mathrm{E}\,L' + 4) + \frac{2}{5}\cdot(\mathrm{E}\,L' + 2)$ and find that $\mathrm{E}\,L = 14$. The *dontGoBack* option complicates the state graph considerably, but we can still compute the expected route length by solving this system of equations for $\mathrm{E}\,L$:

$$\begin{aligned}
\mathrm{E}\,L &= \tfrac{1}{1}\cdot 1 + \mathrm{E}\,L^{ca} \\
\mathrm{E}\,L^{ca} &= \tfrac{1}{4}\cdot 1 + \tfrac{2}{4}\cdot(\mathrm{E}\,L^{co} + 4) + \tfrac{1}{4}\cdot(\mathrm{E}\,L^{lg} + 2) \\
\mathrm{E}\,L^{lg} &= \tfrac{1}{4}\cdot 1 + \tfrac{2}{4}\cdot(\mathrm{E}\,L^{co} + 4) + \tfrac{1}{4}\cdot(\mathrm{E}\,L^{ca} + 2) \\
\mathrm{E}\,L^{co} &= \tfrac{1}{4}\cdot 1 + \tfrac{1}{4}\cdot(\mathrm{E}\,L^{co} + 4) + \tfrac{1}{4}\cdot(\mathrm{E}\,L^{ca} + 2) + \tfrac{1}{4}\cdot(\mathrm{E}\,L^{lg} + 2)
\end{aligned}$$

$\mathrm{E}\,L^{ca}$ is the expected route length **hall**→**kiosk** if the wayfinder just arrived at **hall** from **cafe**, $\mathrm{E}\,L^{lg}$ the same assuming that the wayfinder just arrived at **hall** from the landing **lg**, and $\mathrm{E}\,L^{co}$ if the wayfinder just arrived from the corridor (**c1**,**c2**,**c3**). The result is $\mathrm{E}\,L = 11.2$, which nicely matches the empirical result of 11.46.

This result can also be determined analytically: The particular change to the model results in a longer "Hall-Corridor-Loop." Once the agent goes from the Hall into the Corridor (for example, through **aw2** into **c1**), it has to go through three more links (**u23**, **u13**, and **aw1**) before having a real choice for the next time! Since the agent does not recognise loops it may take this loop several times. Figure 6.4 shows all possible paths and computes the expected route length.

The routes in the alternative model are significantly longer than in the original model, even though the complexity of the environment did not change: while the average number of unique choices increased from $\bar{U} = 2.95$ to $\bar{U} = 7.38$, the average number of random choices increased only from $\bar{R} = 2.95$ to $\bar{R} = 4.08$. These results motivate further research: a measure of complexity for specific routes and for complete Schematic Geometry models.

Another concern might be the optional consistency rules of Schematic Geometry. It is easy to argue for the optional consistency rules, especially for C4 (non-linking elements have at most one immediate predecessor) if Schematic Geometry shall be used as a model of space. But as a model of knowledge of space, which is likely incomplete, the optional consistency rules do not allow for enough generality. It is for this reason that they were declared optional. The Enge and Wiedikon models (Figures 5.3 and 5.4), however, adhere to C3 and C4.

Comparison with Lynch and Alexander. The use of a few schemata as primitives in a model of space and spatial knowledge prompts a comparison with Lynch's five elements [77] and Alexander's pattern language [2].

Lynch's five elements were derived from an empirical study about the "legibility" of three U.S. cities. Except for the Lynchian *path*, which obviously corresponds to the PATH image schema, there is no simple relation between Lynch's elements and image schemata. Lynchian *districts* best correspond to **REGION** and **AGGREGATE** schemata for both their level of scale and the recognisability from within [77:66–72], but there is no obvious correspondence with any one of the image schemata from Johnson's list [54:126]. The real difference between image schemata and Lynch's elements is probably that the former are primarily structural and can be metaphorically projected into symbolic domains, whereas Lynch's elements are of symbolic value but can eventually also be used for structuring (the image of a city).

Alexander defines about 250 *patterns* ranging from "Independent Regions" (pattern 1) to "Ornament" (pattern 249). In contrast to Lynch, Alexander explicitly states the structure of those patterns, a partial order, but the meaning of the ordering relation is left vague[21] and so are the patterns themselves. The success of his pattern language is evidence that this lack of rigour need not be a shortcoming, at least not for practical use.

Schematic Geometry explicitly defines its primitives in terms of image schemata, analyses the resulting structure and finds it to be a partially ordered set. This formal approach allows implementation and the computational wayfinding simulation from Chapter 5. Image schemata, the foundation of Schematic Geometry, are believed to be universal among societies; consequently, Schematic Geometry should be quite generally applicable.

[21] Compare with his earlier work on the structure of cities [1].

Potential weaknesses. As all models do, Schematic Geometry abstracts from the real world. In particular, Schematic Geometry has no geometry, at least not in a Euclidean sense. The implication is that Schematic Geometry *per se* cannot be used for decisions based on distance or direction. This makes a "maintain direction" heuristic or even a "sense of direction" heuristic impossible without extending the model.

Schematic Geometry has no boundaries (apart from those implicit in CONTAINER elements). It is therefore not possible to narrow down the search space by specifying one or the other side of a boundary (like "on the other side of the tracks"). Is that a serious omission? Lynch found boundaries (he called them edges) to be important elements in cognitive maps [77] and Kuipers showed how boundaries can be used for spatial reasoning [65]. However, spatial reasoning with boundaries always identifies a region and if this region can be immediately identified, then there is no need for reasoning with boundaries. In the two example environments of Chapter 5, there are no regions that could not be directly identified; boundaries (like the tramway tracks) only serve to cut the station square into two pieces, which is not of much use because the square is just one scene and thus a coherent unit. When considering other types of environments, especially entire cities, reasoning with boundaries is important and useful.

Vision is the main source of wayfinding information but visibility is only implicitly present in Schematic Geometry, namely in the definition of scenes. It is based on the presumably opaque boundary of CONTAINERS: REGIONS are assumed wholly visible from within, and the inside of CONTAINERS is assumed to be occluded from the outside. This is certainly a simplification, and counter-examples are not hard to find, like a non-convex REGION or a CONTAINER with a transparent boundary.Nevertheless, with the two stations examined in Chapter 5, this simple model matches reality fairly well. If more detail with respect to visibility is desired, it could be added (a) by breaking REGIONS into convex parts and connecting them with ULINKS and (b) by using attributes to mark CONTAINERS as transparent or using previews. Of course, the wayfinding agent has to be configured to use this information.

Chapter 7

Conclusions and Outlook

Schematic Geometry is developed as a model of the structure of scene space. It builds on image schemata, mental patterns that help us making sense of our environment by structuring it. This makes Schematic Geometry both an abstraction and an enrichment of the environment. On the one hand, it is a discretisation of intrinsically continuous space; on the other hand, it explicitly represents spatial structure that is only implicitly present in the environment.

A software agent, called the "standard wayfinder," is designed to find its way using only structural information represented by Schematic Geometry. The agent is tested in Schematic Geometry models of two small railway stations with non-trivial layouts. Results of this simulation provide a detailed picture of the need for different types of information necessary for successful wayfinding.

7.1 Conclusions

At the outset of this thesis, two types of environments for wayfinding in public transport systems are distinguished: those that exhibit a clear network structure, called *network space*, and those without an obvious network structure, called *scene space*. This classification is a **premise** for the present work and no prior research was done to justify it. Properties of the two types are analysed with respect to spatial reasoning, and compared to the relevant literature.

This research suggests that wayfinding in scene space is not fundamentally different from wayfinding in network space, as another kind of a network *emerges* from scene space. This network is called the *scene graph*. It is not as obvious as in network space, but results from an analysis of the environment. A closer look at scene space reveals a hierarchical structure. This hierarchy illustrates which part of the environment belongs to which other part, thus

providing useful global information for local decisions during wayfinding. In network space (lacking such a "super-structure"), there is no global information available at the individual "decision points" (i.e., nodes in the network), and therefore signage is vital for successful wayfinding in network space.

The fundamental concept on which this thesis is built are *image schemata*, cognitive structural patterns. In a simple example, like "the pawn is on the chess board," the structuring power of image schemata is obvious. This thesis postulates that image schemata can also be applied to structure complete spatial configurations, like railway stations, and help humans to "make sense" of space while wayfinding (**hypothesis 1**).

The successful application of Schematic Geometry to the modelling of two railway stations shows that image schemata can be used to represent complete spatial configurations. It is interesting to analyse the resulting *formal* structure, which turns out to be a partial ordering. To refine the proposed model, additional image schemata can be considered such as VERTICALITY (for up/down relations) or even ATTRACTION (to express the salience of an object). Image schemata enjoyed great popularity in GIScience in the 1990ies but then interest waned. This work might suggest a "renaissance."

Concerning wayfinding, **hypothesis 2a** posits that structural information suffices to guide a traveller through scene space. Results show that this is, in general, false: the agent often finds a non-optimal way. This behaviour is caused by structurally equivalent choices and lacking survey knowledge. **Hypothesis 2b** claims that signage and previews account for this difference between actual and optimal wayfinding. The experimental results give no evidence that this is false. Rather, it hints that in many cases only a few well placed signs (or better: previews) enable the wayfinder to perform optimally.

While this shows that structural information *alone* is not sufficient for efficient wayfinding, structure and especially hierarchical structure is still essential to wayfinding, because it provides a framework for efficient signage. Therefore, considerations about wayfinding should go along with the design of an architectural layout, not only with signage at a later stage.

Achievements. First, a distinction between network space and scene space characterises public transport *as an environment for wayfinding*. The traveller using public transport is exposed to two significantly different environments, which is not the case in private car transportation. Public transport operators have to be aware of this and provide the necessary information for both types in suitable places and forms.

Second, a model for scene space, called Schematic Geometry, was developed. It trades cognitive relevance for geometric precision. The strong cognitive foundation ensures that the model captures what is actually meaningful to human reasoning, though it does not necessarily capture *all* that is meaningful; in particular, there are researchers who believe that metrical information (not represented in Schematic Geometry) is important to human wayfinding [88].

Third, environmental structures that support wayfinding are identified. Success and failure of the wayfinding agent indicate (*i*) that hierarchical structure is only useful if it helps distinguish gateways on the level of the scene graph, (*ii*) that previous knowledge is an important frame into which locally observed structure will be integrated, and (*iii*) that previews are useful to make distant places available for local decision making. This hints at a worthwhile field for further research: the development of measures that compactly express the complexity of an environment, and methods to predict the efficiency of a wayfinding method (such as structure-based or preview-based) in a given environment without the need for simulation.

Contribution. This thesis (*i*) introduces a novel, structure-based classification of environments, namely *network space* and *scene space*, (*ii*) it contributes to our understanding of wayfinding in scene space (environments without a clear network structure), and (*iii*) it provides a qualitative, cognitively motivated model for scene space environments, *Schematic Geometry*. This is of importance to cognitive science (especially spatial cognition), geographic information science, architecture and transportation.

Recommendations for practice. This research provides scientific support for the following design recommendations for public transport interchange nodes: (1) Design for previews, they are more universally understandable than signs; previews are of most use if they identify a gateway as this allows for immediate decision-making. (2) Adherence to a common structural pattern allows travellers to benefit maximally from previous knowledge; this is especially true of the "high level" structure and also helps visually impaired people. (3) Build direct links between all "high level" elements like the station square, the main building, and the platform area, as this makes the *focusSearch* heuristic most effective. If this is not possible, previews are a good substitute. (4) Make global information locally available: this allows for robust—and therefore comfortable—wayfinding.

7.2 Outlook

Wayfinding draws on information from many different sources. Signage is the best known source, but many others exist, including landmarks, gradients, and the cognitive map. This work looked at spatial structure as an information source and other works looked at other types of information sources. What remains to be done is a comprehensive study of the interaction between information from different sources, how one source can compensate for the lack of another source, how humans integrate information from various sources, and how they deal with inconsistencies. This is a major project.

More manageable projects that are closely linked to the present thesis are described next: extending Schematic Geometry, adapting it for modelling wayfinding at the urban level, and studying applications beyond the simulation of wayfinding.

Extending Schematic Geometry. There are situations (not examined in the two station examples earlier) when it is not clear if a link qualifies as a GATEWAY or a ULINK. What is really needed is a GATEWAY in one direction (from a wider place to a narrower place) and a ULINK in the other direction (from narrow to wide). While this extension does not change the partonomic structure, its specific representation has to be studied.

It is also desirable to make the model more dynamic in accordance with the real environment. For example, when boarding or leaving a train there are many GATEWAYS between the train and the platform. Moreover, there are often crowds moving around, so a CROWD schema (probably a complicated superimposition of Johnson's ATTRACTION, BLOCKAGE, and COMPULSION image schemata) might be considered as well. Again, the wayfinding agent has to be adapted accordingly.

Speaking of the agent, its learning skills should be improved, especially route learning to avoid repeatedly walking the same loop. This requires a careful study of the learning related literature but is expected to improve the agent's performance (route length) considerably.

Finally, Schematic Geometry and the standard wayfinder's reasoning should be linked with a classical locomotion model (like PEDFLOW [59]) because they complement each other almost perfectly: Locomotion models handle the details of obstacle avoidance, crowd behaviour, and maintenance of a direction, but are not concerned with strategic planning and decision-making.

The urban level. Schematic Geometry is designed specifically for railway stations by its choice of cognitive spatial schemata. It seems particularly interesting to generalise Schematic Geometry to the larger urban level, where network space and scene space aspects are about equally prominent. New schemata are needed to represent street segments. They will likely build up on the PATH image schema. A boundary representation seems necessary as well, but it is not clear if they should be proper schemata or derived features. Would those changes to Schematic Geometry also change the fundamental partonomic structure? I hypothesise that it will not change, but that further use of attributes will be required; a careful structural analysis will be necessary here. Another approach would be to start with Lynch's elements at the foundation (instead of using cognitive spatial schemata) and analysing what formal structure results from their relational properties.

Schematic Geometry beyond wayfinding. Schematic Geometry was developed to simulate wayfinding, but it can also serve other purposes.

For example, Schematic Geometry can be used as a tool for *structural analysis* of environments, which is relevant for architecture and landscape planning. From an identification of cognitive spatial schemata and their interrelation, it comptues scenes, finds the scene graph, and tells us about the hierarchical structure. However, further research is needed to answer questions like: "how similar are the Enge and the Wiedikon railway station?" or: "is the station in Figure 4.8 a common subset of Enge and Wiedikon?" These questions are related with finding graph morphisms (structure-preserving mappings) between two Schematic Geometries. For example, the station in Figure 4.8 can almost be mapped into the Enge station (Figure 5.3) by the morphism { **pa**↦**pa**, **p1**↦**p1**, **p2**↦**p2**, **sub**↦**un**, **bg**↦**bg**, **hall**↦**cr**, **sq**↦**sq**, **trs**↦**t567**, **pe**↦**do**, **me**↦**me** }; "almost" because there is no place where to map **ul**: this direct connection between the Square and Platform 1 does not exist at the Enge station. This is a significant structural difference. In a similar way, the assumed general knowledge about a railway station (Figure 5.7) can be mapped into the Enge and the Wiedikon station, so it is a "common core" of the two stations. But in how many ways can it be mapped? And which parts of the stations are not covered by the mapped image of the general knowledge? While it is easy to find such mappings by hand (or realising that there is no such mapping) for simple cases, in general, this is a very difficult task and the author is not aware of an efficient algorithm. To further complicate things, it is not "simply" a matter of finding a graph morphism, because schemata also have to be taken into account: for example, we cannot map a GATEWAY into a REGION, but it can be discussed if

it shall be allowed to map a REGION into an AGGREGATE because an AGGREGATE essentially generalises the notion of a REGION. Analyses of this kind can be used for a comprehensive assessment of a station's conformance with design and complexity guidelines, but requires a significant research effort to work out the details.

Another fruitful area for further research is the *definition of measures* that (*i*) succinctly characterise the structure of a railway station or any other scene space, (*ii*) summarise the structural importance of a particular element, or (*iii*) express the complexity of a specific route. This research will certainly draw on the existing measures for graphs (see [42:32] for a geographically oriented list) but must also take the hierarchical structure into account. For example, the GATEWAY **do** at Enge not only connects the Circular Room with Platform 1, but also the Building with the entire Platform Area (inheritance of location). This may be considered more important than the GATEWAY **co**, which "only" connects two parts within the building. One can use depth in the dominator tree (which arises naturally from the notion of *dominance*, see page 51) to express a gateway's importance: **do** has depth 1, whereas **co** has depth 3, and generally, the deeper a GATEWAY, the less important it is.

Finally, it is tempting to *generate route descriptions* from a Schematic Geometry. Given the linguistic background of image schemata, this temptation seems obvious. The overall procedure would be to first find an optimal route (with respect to a user profile that may include things like "avoid stairs"), then to use the underlying image schemata to describe this route. For example, the route **t567-sq-me-cr-do-p1** at Enge could be described as

1. through the *main entrance* into the *station building*,
2. you are now in a *circular room*,
3. leave the *building* (and the room) through the *glass doors*,
4. you are now on Platform 1.

This route description contains some redundancy, which serves as confirmatory information and therefore is generally desirable (an open question is how much redundancy is desirable). Underlined words were derived from the schemata: because the main entrance is a GATEWAY we go through it and because the building is a CONTAINER we go into it (but end up on Platform 1, a REGION; image schemata specify the generic *part-of* relation). Words in italics must be derived from attributes (such as an element's "name" attribute) because there is no appropriate schematic information. Moreover, attribute information has to be used wherever the standard wayfinder would have to make a random decision, as is the case with the Enge station square, where there are three structurally equivalent gateways into the building. Such

places are likely to be places where Norman's "knowledge in the world" and "knowledge in the head" [93] should be supplemented by "knowledge in the pocket" [122]. In this way, Schematic Geometry is also useful as a model for anticipating and answering queries for a location-based service.

Darest thou now, O Soul,
Walk out with me toward the Unknown Region,
Where neither ground is for the feet, nor any path to follow?

No map, there, nor guide,
Nor voice sounding, nor touch of human hand,
Nor face with blooming flesh, nor lips, nor eyes, are in that land.

I know it not, O Soul;
Nor dost thou – all is a blank before us;
All waits, undreamed of, in that region – that inaccessible land.

Till, when the ties loosen,
All but the ties eternal, Time and Space,
Nor darkness, gravitation, sense, nor any bounds, bounding us.

Then we burst forth, we float
In Time and Space, O Soul, prepared for them;
Equal, equipt at last, (O joy! O fruit of all!) them to fulfil, O Soul.

—Walt Whitman (1819–1892)

Chapter 8

References

[1] C. Alexander, "A city is not a tree", *Architectural Forum* **122** (1965).

[2] C. Alexander, S. Ishikawa & M. Silverstein, *A Pattern Language: towns, buildings, construction*, Oxford University Press, New York, 1977.

[3] G. L. Allen, "Spatial Abilities, Cognitive Maps, and Wayfinding", in *Wayfinding Behavior* (Chapter 2), R. G. Golledge, ed., Johns Hopkins University Press, Baltimore, Maryland, 1999, 46–80.

[4] P. Arthur & R. Passini, *Wayfinding: People, Signs, and Architecture*, McGraw-Hill, New York, 1992.

[5] T. Barkowsky, "Mental Representation and Processing of Geographic Knowledge", University of Hamburg (Germany), Ph.D. thesis, 2002.

[6] M. L. Benedikt, "To take hold of space: isovists and isovist fields", *Environment and Planning B* **6** (1979), 47–65.

[7] M. Boesch, "Engagierte Geographie. Zur Rekonstruktion der Raumwissenschaft als Politikorientierte Geographie", *Erdkundliches Wissen* **98** (1989).

[8] P. H. L. Bovy & E. Stern, *Route Choice: Wayfinding in Transport Networks*, Kluwer Academic Publishers, Dordrecht, 1990.

[9] H. Brändli, "Angebote des öffentlichen Verkehrs", Swiss Federal Institute of Technology, Scriptum IVT ETHZ, Zürich, 1984.

[10] H. Brändli, "Konfliktstellen Individualverkehr ⟷ Öffentlicher Linienverkehr", Swiss Federal Institute of Technology, IVT draft technical report, Zürich, 2001.

[11] C. M. Brugman, "The story of 'over'", University of California, Berkely, M.A. thesis, 1981, reproduced by the Indiana University Linguistics Club in 1983.

[12] D. Caduff, "Assessing Landmark Salience for Wayfinding Tasks", University of Zürich, Ph.D. thesis (in progress).

[13] C. H. Cap, *Theoretische Grundlagen der Informatik*, Springer-Verlag, Wien, 1993.

[14] A. Car, "Hierarchical Spatial Reasoning. Theoretical Consideration and its Application to Modeling Wayfinding", Technical University Vienna, Ph.D. thesis, Vienna, Austria, 1996.

[15] E. Chown, "Error Tolerance and Generalization in Cognitive Maps. Performance Without Precision", in *Wayfinding Behavior*, R. G. Golledge, ed., Johns Hopkins University Press, Baltimore, Maryland, 1999, 349–369.

[16] E. Chown, "Gateways: An approach to parsing spatial domains", *ICML 2000 Workshop on Machine Learning of Spatial Knowledge*, 2000.

[17] E. Chown, S. Kaplan & D. Kortenkamp, "Prototypes, Location, and Associative Networks (PLAN): Towards a Unified Theory of Cognitive Mapping", *Cognitive Science* **19** (1995), 1–51.

[18] W. J. Clancey, *Situated Cognition*, Cambridge University Press, Cambridge, UK, 1997.

[19] A. G. Cohn, "The Challenge of Qualitative Spatial Reasoning", *ACM Computing Surveys* **27** (1995), 323–325.

[20] K. D. Cooper, T. J. Harvey & K. Kennedy, "A Simple, Fast Dominance Algorithm", *Software Practice and Experience* **4** (2001), 1–10.

[21] R. P. Darken, T. Allard & L. B. Achille, "Spatial Orientation and Wayfinding in Large-Scale Virtual Spaces", *Presence: Teleoperators and Virtual Environments* **8** (1999), iii–vi.

[22] F. Detje, D. Dörner & H. Schaub, eds., *The Logic of Cognitive Systems* Proceedings of the 5th Intl Conf on Cognitive Modeling ICCM-5, Institut für Theoretische Psychologie, Universität Bamberg, Bamberg, Germany, 2003.

[23] R. B. Dewell, "Over again: Image-schema transformations in semantic analysis", *Cognitive Linguistics* **5** (1994), 351–380.

[24] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* **1**, 269–271 & 1959.

[25] R. M. Downs & D. Stea, *Maps in Minds: Reflections on cognitive mapping*, Harper & Row, New York, 1977.

[26] B. Dushnik & E. W. Miller, "Partially ordered sets", *American Journal of Mathematics* **63** (1941), 600–610.

[27] S. I. Fabrikant & A. Skupin, "Cognitively Plausible Information Visualization", in *Exploring GeoVisualization*, Elsevier, Amsterdam, 2005, 667–690.

[28] A. Fall, "Reasoning with taxonomies", Simon Fraser University, Ph.D. thesis, Burnaby, British Columbia, Canada, 1996.

[29] A. U. Frank & M. Raubal, "Formal Specification of Image Schemata. A Step to Interoperability in Geographic Information Systems", *Spatial Cognition and Computation* **1** (1999), 67–101.

[30] C. Freksa, "Qualitative Spatial Reasoning", in *Cognitive and Linguistic Aspects of Geographic Space*, D. M. Mark & A. U. Frank, eds., Kluwer Academic, Dordrecht, The Netherlands, 1991, 361–372.

[31] S. M. Freundschuh & M. J. Egenhofer, "Human Conceptions of Spaces: Implications for Geographic Information Systems", *Transactions in GIS* **2** (1997), 361–375.

[32] S. M. Freundschuh & M. Sharma, "Spatial Image Schemata, Locative Terms, and Geographic Spaces in Children's Narrative: Fostering Spatial Skills in Children", *Cartographica* **32** (1996), 38–49.

[33] A. Galton, "Lines of Sight", *AI and Cognitive Science*, 1994, 103–113.

[34] T. Gärling, A. Böök & E. Lindberg, "Spatial Orientation and Wayfinding in the Designed Environment. A conceptual analysis and some suggestions for postoccupancy evaluation", *Journal of Architectural and Planning Research* **3** (1986), 55–64.

[35] Z. Genç, "Ein neuer Ansatz zur Fahrplanoptimierung im ÖPNV: Maximierung von zeitlichen Sicherheitsabständen", University of Cologne, Ph.D. thesis, 2003.

[36] J. J. Gibson, *The Ecological Approach to Visual Perception*, Lawrence Erlbaum Associates, London, 1979.

[37] M. Gluck, "Making Sense of Human Wayfinding: Review of cognitive and linguistic knowledge for personal navigation with a new research direction", in *Cognitive and Linguistic Aspects of Geographic Space*, Kluwer Academic Press, Dordrecht, The Netherlands, 1991, 117–135.

[38] R. G. Golledge, "Time and Space in Route Preference", University of California Transportation Center, Working Paper UCTC No. 213, 1993.

[39] R. G. Golledge, "Path Selection and Route Preference in Human Navigation: A Progress Report", *Lecture Notes in Computer Science* **988** (1995), 207–222, (Proc *COSIT*).

[40] R. G. Golledge, ed., *Wayfinding Behavior*, The Johns Hopkins University Press, Baltimore, Maryland, 1999.

[41] S. Gopal, R. Klatzky & T. Smith, "NAVIGATOR: A Psychologically Based Model of Environmental Learning Through Navigation", *Journal of Environmental Psychology* **9** (1989), 309–331.

[42] P. Haggett & R. J. Chorley, *Network Analysis in Geography*, Edward Arnold, London, 1969.

[43] S. Harnad, "The symbol grounding problem", *Physica D* **42** (1990), 335–346.

[44] J. L. Hein, *Discrete Mathematics* (second edition), Jones and Bartlett Publishers, Sudbury, Massachusetts, 2003.

[45] C. Heye, U-J. Rüetschi & S. Timpf, "Komplexität von Routen in öffentlichen Verkehrssystemen", *Angewandte Geographische Informationsverarbeitung XV (AGIT)*, Salzburg, Austria, 2003, 159–168.

[46] L. L. Hill, S. J. Crosier, T. R. Smith & M. Goodchild, "A Content Standard for Computational Models", *D-Lib Magazine* **7** (2001).

[47] B. Hillier & J. Hanson, *The Social Logic of Space*, Cambridge University Press, 1984.

[48] S. C. Hirtle, "The Cognitive Atlas: Using GIS as a Metaphor for Memory", in *Spatial and temporal reasoning in geographic information systems*, Oxford University Press, 1998, 263–271.

[49] S. C. Hirtle & J. Jonides, "Evidence of hierarchies in cognitives maps", *Memory & Cognition* **13** (1985), 208–217.

[50] H. H. Hochmair & A. U. Frank, "Influence of estimation errors on wayfinding-decisions in unknown street networks – analyzing the least-angle strategy", *Spatial Cognition and Computation* **2** (2002), 281–313.

[51] E. Hutchins, *Cognition in the Wild*, M.I.T. Press, Cambridge, Massachusetts, 1995.

[52] Infopolis 2 Consortium, "Needs of travellers: An Analysis Based on the Study of Their Tasks and Activities", Commission of the European Communities, DG XIII, TR 4016, Brussels, 1999.

[53] T. Ishikawa, "Spatial Knowledge Acquisition in the Environment: The Integration of Separately Learned Places and the Development of Metric Knowledge", University of California, Santa Barbara, Ph.D. thesis, 2002.

[54] M. Johnson, *The Body in the Mind. The Bodily Basis of Meaning, Imagination, and Reason*, University of Chicago Press, Chicago, 1987.

[55] C. Joslyn, "Poset Ontologies and Concept Lattices as Semantic Hierarchies", *Lecture Notes in Artificial Intelligence* **3127** (2004), 287–302, (Proc Intl Conf on *Conceptual Structures*).

[56] W. Kainz, "Application of Lattice Theory to Geography", *Proc 3rd Intl Symp on Spatial Data Handling*, Sydney, 1988, 135–142.

[57] W. Kainz, "Spatial Relationships: Topology versus Order", *Proc 4th Intl Symp on Spatial Data Handling*, Zürich, 1990, 814–819.

[58] W. Kainz, M. J. Egenhofer & I. Greasly, "Modelling Spatial Relations and Operations with Partially Ordered Sets", *International Journal of Geographical Information Systems* **7** (1993), 214–229.

[59] J. Kerridge, J. Hine & M. Wigan, "Agent-based modelling of pedestrian movements: the questions that need to be asked and answered", *Environment and Planning B: Planning and Design* **28** (2001), 327–341.

[60] R. M. Kitchin, "Cognitive maps: What are they and why study them?", *Journal of Environmental Psychology* **14** (1994), 1–19.

[61] A. Klippel, "Wayfinding Choremes: Conceptualizing Wayfinding and Route Direction Elements", University of Bremen, Ph.D. thesis, 2003.

[62] D. E. Knuth, *Fundamental Algorithms* (third edition), The Art of Computer Programming #**1**, Addison-Wesley, Reading, Massachusetts, 1997.

[63] W. Kuhn & A. U. Frank, "A Formalization of Metaphors and Image-Schemas in User Interfaces", in *Cognitive and Linguistic Aspects of Geographic Space*, D. M. Mark & A. U. Frank, eds., Kluwer Academic Publishers, 1991, 419–434.

[64] B. J. Kuipers, "Representing Knowledge of Large-Scale Space", Massachusetts Institute of Technology, Ph.D. thesis, 1977.

[65] B. J. Kuipers, "Modeling Spatial Knowledge", *Cognitive Science* **2** (1978), 129–153.

[66] B. J. Kuipers, "On Representing Commonsene Knowledge", in *Associative Networks: The Representation and Use of Knowledge by Computers*, N. V. Findler, ed., Academic Press, New York, 1979, 393–408.

[67] B. J. Kuipers, "The "map in the head" metaphor", *Environment and Behavior* **14** (1982), 202–220.

[68] B. J. Kuipers, "The Cognitive Map: Could It Have Been Any Other Way?", in *Spatial Orientation: Theory, Research, and Application*, H. L. Pick & L. P. Acredolo, eds., Plenum Press, New York, 1983, 345–359.

[69] B. J. Kuipers, "The Spatial Semantic Hierarchy", *Artificial Intelligence* **119** (2000), 191–233.

[70] B. J. Kuipers, "Interview", *Künstliche Intelligenz* **2002** (2002), 40.

[71] M-P. Kwan & J. Weber, "Individual Accessibility Revisited: Implications for Geographical Analysis in the Twenty-First Century", *Geographical Analysis* **35** (2003), 341–353.

[72] G. Lakoff, *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind*, University of Chicago Press, Chicago, 1987.

[73] R. Langacker, *Foundations of Cognitive Grammar* (volume 1), Stanford University Press, 1987.

[74] D. Leiser & A. Zilbershatz, "The traveller: A computational model of spatial network learning", *Environment and Behavior* **21** (1989), 435–463.

[75] R. Lloyd, R. Cammack & W. Holliday, "Learning Environments and Switching Perspectives", *Cartographica* **32** (1996), 5–17.

[76] P. Lüthi, "Bahnhof Enge, Zürich (1925–1927)", Zürcher Hochschule Winterthur, Architekturabteilung, 2000, KGS-Semesterarbeit.

[77] K. Lynch, *The Image of the City*, M.I.T. Press, Cambridge, Massachusetts, 1960.

[78] J. Maier & H-D. Atzkern, *Verkehrsgeographie*, Teubner, Stuttgart, 1992.

[79] D. M. Mark & A. U. Frank, "Experiential and Formal Models of Geographic Space", *Environment and Planning B* **23** (1996), 3–24.

[80] D. M. Mark, A. U. Frank, M. J. Egenhofer, S. M. Freundschuh, M. McGranaghan & R. M. White, "Languages of Spatial Relations: Initiative 2 Specialist Meeting Report", NCGIA, TR 89-2, 1989.

[81] G. Mazzola, *The Topos of Music*, Birkhäuser, Basel and Boston, 2002.

[82] G. McCalla, L. Reid & P. Schneider, "Plan Creation, Plan Execution, and Knowledge Acquisition in a Dynamic Microworld", *International Journal of Man-Machine Studies* **16** (1982), 89–112.

[83] T. McNamara, "Mental representations of spatial relations", *Cognitive Psychology* **18** (1986), 87–121.

[84] G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processsing Information", *Psychological Review* **63** (1956), 81–97.

[85] G. A. Miller, "The cognitive revolution: a historical perspective", *TRENDS in Cognitive Sciences* **7** (2003), 141–144.

[86] D. R. Montello, "Scale and Multiple Psychologies of Space", *Lecture Notes in Computer Science* **716** (1993), 312–321, (Proc *COSIT*).

[87] D. R. Montello, "A new framework for understanding the acquisition of spatial knowledge in large-scale environments", in *Spatial and temporal reasoning in geographic information systems*, M. J. Egenhofer & R. G. Golledge, eds., Oxford University Press, 1998, 143–154.

[88] D. R. Montello, "Navigation", in *The Cambridge handbook of visuospatial thinking*, P. Shah & A. Miyake, eds., Cambridge University Press, Cambridge, UK, 2005, 257–294.

[89] J. Muhlhausen, "Wayfinding Is Not Signage", *Signs of the Times magazine*, 2006.

[90] U. Neisser, *Cognition and Reality. Principles and Implications of Cognitive Psychology*, Freeman, New York, 1976.

[91] G. Nelson, *The Inform Designer's Manual* (4th edition), The Interactive Fiction Library, St. Charles, Illinois, 2001.

[92] R. van Nes, "Hierarchical network levels in the design of multimodal transport networks", *NECTAR Conference*, Delft, The Netherlands, 1999.

[93] D. A. Norman, *The design of everyday things*, Doubleday, New York, 1988.

[94] K. Oatley, "Mental maps for navigation", *New Scientist* **19** (1974), 863–866.

[95] O. Pearce, "Verfügbarkeit des öffentlichen Verkehrs in Zürich zu verschiedenen Tageszeiten", University of Zürich, Diploma thesis, 2006.

[96] O. Pearce & S. Timpf, "Erreichbarkeit von Haltestellen des öffentlichen Verkehrs zu verschiedenen Tageszeiten", *18. Symposium Angewandte Geoinformatik (AGIT)*, Salzburg, Austria, 2006, 535–544.

[97] R. Pfeifer & C. Scheier, *Understanding Intelligence*, Bradford Books, M.I.T. Press, Cambridge, Massachusetts, 1999.

[98] J. Portugali & I. Omer, "Systematic Distortions in Cognitive Maps: The North American West Coast vs. the (West) Coast of Israel", *Lecture Notes in Computer Science* **2825** (2003), 93–100, (Proc *COSIT*).

[99] C. C. Presson, N. DeLange & M. Hazelrigg, "Orientation specificity in spatial memory: What makes a path different from a map of the path?", *Journal of Experimental Psychology: Learning, Memory, and Cognition* **15** (1989), 887–897.

[100] M. Raubal, "Agent-based simulation of human wayfinding: A perceptual model for unfamiliar buildings", Technical University Vienna, Ph.D. thesis, 2001.

[101] M. Raubal, ed., *Wayfinding in Built Environments. The Case of Airports*, IfGI prints #**14**, Institut für Geoinformatik, Münster, Germany, 2002.

[102] M. Raubal & M. J. Egenhofer, "Comparing the complexity of wayfinding tasks in built environments", *Environment & Planning B: Planning and Design* **25** (1998), 895–913.

[103] M. Raubal, M. J. Egenhofer, D. Pfoser & N. Tryfona, "Structuring Space with Image Schemata: Wayfinding in Airports as a Case Study", *Lecture Notes in Computer Science* **1329** (1997), 85–102, (Proc *COSIT*).

[104] M. Raubal & M. F. Worboys, "A Formal Model of the Process of Wayfinding in Built Environments", *Lecture Notes in Computer Science* **1661** (1999), 381–399, (Proc *COSIT*).

[105] A. M. Rodriguez & M. J. Egenhofer, "Image-Schemata-Based Spatial Inferences: The Container-Surface Algebra", *Lecture Notes in Computer Science* **1329** (1997), 35–52, (Proc *COSIT*).

[106] U-J. Rüetschi & S. Timpf, "Schematic Geometry of Transport Spaces for Wayfinding", *IfGI prints* **22** (2004), 191–203, (Proc *Münsteraner GI-Tage: Geoinformation und Mobilität*).

[107] U-J. Rüetschi & S. Timpf, "Modelling Wayfinding in Public Transport: Network Space and Scene Space", *Lecture Notes in Artificial Intelligence* **3343** (2005), 24–41, (Proc Intl Conf *Spatial Cognition IV*).

[108] U-J. Rüetschi & S. Timpf, "Using Image Schemata to Represent Meaningful Spatial Configurations", *Lecture Notes in Computer Science* **3762** (2005), 1047–1055, (Proc Intl Workshop *Semantic-based GIS*).

[109] U-J. Rüetschi & S. Timpf, "Cognitively Motivated Simulation of Transfers in Public Transport", *International Symposium on Transport Simulation*, Lausanne, Switzerland, 2006.

[110] D. E. Rumelhart, "The Room Theory", University of California, San Diego, unpublished computer listing, La Jolla, 1974.

[111] S. J. Russel & P. Norvig, *Artificial Intelligence. A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[112] B. S. W. Schröder, *Ordered Sets*, Birkhäuser, Boston, 2003.

[113] B. Shanon, "Where Questions", *Association for Computational Linguistics (Proc 17th annual meeting)*, La Jolla, California, 1979, 73–75.

[114] A. W. Siegel & S. H. White, "The Development of Spatial Representations of Large-Scale Environments", in *Advances in Child Development and Behaviour*, H. W. Reese, ed., Academic Press, New York, 1975, 9–55.

[115] H. A. Simon, "The architecture of complexity", in *The sciences of the artificial*, H. A. Simon, ed., M.I.T. Press, Cambridge, Massachusetts, 1969, 192–229.

[116] M. E. Sorrows & S. C. Hirtle, "The Nature of Landmarks for Real and Electronic Spaces", *Lecture Notes in Computer Science* **1661** (1999), 37–50, (Proc *COSIT*).

[117] H. Stachowiak, *Allgemeine Modelltheorie*, Springer-Verlag, Wien, 1973.

[118] A. Stevens & P. Coupe, "Distortions in judged spatial relations", *Cognitive Psychology* **10** (1978), 422–437.

[119] Q. Stevens, "The shape of urban experience: a reevaluation of Lynch's five elements", *Environment & Planning B: Planning and Design* **33** (2006), 803–823.

[120] L. Talmy, "How language structures space", in *Spatial Orientation: Theory, Research, and Application*, H. Pick & L. Acredolo, eds., Plenum Press, New York, 1983, 225–282.

[121] S. Timpf, "Ontologies of Wayfinding: a Traveler's Perspective", *Networks and Spatial Economics* **2** (2002), 9–22.

[122] S. Timpf, "Wayfinding with mobile devices: decision support for the mobile citizen", in *Frontiers of Geographic Information Technology*, S. Rana & J. Sharma, eds., Springer-Verlag, Berlin, 2005, 209–228.

[123] S. Timpf & W. Kuhn, "Granularity Transformations in Wayfinding", *Lecture Notes in Artificial Intelligence* **2685** (2003), 77–88, (Proc *Spatial Cognition III*).

[124] S. Timpf, G. S. Volta, D. W. Pollock & M. J. Egenhofer, "A Conceptual Model of Wayfinding Using Multiple Levels of Abstractions", *Lecture Notes in Computer Science* **639** (1992), 348–367, Proc Conf *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*.

[125] E. C. Tolman, "Cognitive maps in rats and men", *Psychological Review* **55** (1948), 189–208.

[126] W. T. Trotter, *Combinatorics and partially ordered sets: dimension theory*, Johns Hopkins University Press, Baltimore, 1992.

[127] C. C. Trowbridge, "On fundamental methods of orientation and imaginary maps", *Science* **38** (1913), 888–897.

[128] A. Tversky, "Features of Similarity", *Psychological Review* **84** (1977), 327–352.

[129] B. Tversky, "Distortions in Memory for Maps", *Cognitive Psychology* **13** (1981), 407–433.

[130] B. Tversky, "Distortions in Cognitive Maps", *Geoforum* **23** (1992), 131–138.

[131] B. Tversky, "Cognitive Maps, Cognitive Collages, and Spatial Mental Models", *Lecture Notes in Computer Science* **716** (1993), 14–24, (Proc *COSIT*).

[132] A. Varzi, "Mereology", in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, ed., Stanford University, 2003, http://plato.stanford.edu/archives/sum2003/entries/mereology/.

[133] J. Weisman, "Evaluating architectural legibility: Way-finding in the built environment", *Environment and Behavior* **13** (1981), 189–204.

[134] J. M. Wiener & H. A. Mallot, "'Fine-to-Coarse' Route Planning and Navigation in Regionalized Environments", *Spatial Cognition and Computation* **3** (2003), 331–358.

[135] S. Winter, "Route Specifications with a Linear Dual Graph", *Advances in Spatial Data Handling*, 2002, 329–338, Proc Intl Symp on *Spatial Data Handling*.

[136] M. F. Worboys, *GIS: A computing perspective*, Taylor & Francis, 1995.

[137] M. Yannakakis, "The complexity of the partial order dimension problem", *Algebraic and Discrete Methods* **3** (1982), 351–358.

Appendix A

Partial Ordering

Given a set of elements, a total (or linear) order determines the order between *all* pairs of elements, whereas a partial order only determines the order between *some* pairs of elements. This is useful to represent many concepts such as parthood ("the hall is part of the building"), workflow ("before assembling the car, build the engine"), or dependence ("Chapter 4 depends on Appendix A"). This appendix explains those concepts of partial ordering that are used in this thesis; it is not a complete account on ordering theory.

Formally, a *partially ordered set* or *poset* $(X, \prec)$ consists of an ordering relation $\prec$ over a ground set X such that for all x, y, z in X we have:

$x \prec y$ and $y \prec z$ implies $x \prec z$	transitivity
$x \prec y$ implies $y \not\prec x$	asymmetry
$x \not\prec x$	irreflexivity

The notation $x \prec y$ is pronounced "x precedes y" or "x is less than y" or whatever is appropriate for the real-world concept represented by the poset. The second property is actually superfluous because any relation that is transitive and irreflexive is always asymmetric. To prove, assume $x \prec y$ and $y \prec x$, then by transitivity $x \prec x$, contradicting irreflexivity.

Starting from an ordering relation $\prec$ we can define a new relation $\preceq$ such that $x \preceq y$ whenever $x \prec y$ or $x = y$. For the new relation these properties can be derived:

$x \preceq y$ and $y \preceq z$ implies $x \preceq z$	transitivity
$x \preceq y$ and $y \preceq x$ implies $x = y$	antisymmetry
$x \preceq x$	reflexivity

Proof: Transitivity follows immediately from the definition of $\preceq$ and the transitivity of $\prec$. Whenever $x \preceq y$ and $y \preceq x$ then by the asymmetry of $\prec$ neither $x \prec y$ nor $y \prec x$ can hold, and therefore $x = y$, proving the antisymmetry of $\preceq$. Finally, the reflexivity of $\preceq$ is inherited from the reflexivity of equality.

Using similar reasoning we can derive the first three properties from the latter three if we define $x \prec y$ to mean $x \preceq y$ but $x \neq y$. Therefore, either set of properties can be used to define a partial ordering over a set. Posets of the type $(X, \prec)$ are called *strict* and posets of the type $(X, \preceq)$ are called *reflexive*. This appendix assumes strict posets, unless explicitly stated otherwise.

If the ordering relation is called neither $\prec$ nor $\preceq$, but, say, P, we will nevertheless write $x \prec y$ in P instead of $(x, y) \in P$. Given a poset (X, P), two elements x, y in X with $x \neq y$ are called *comparable*, denoted $x \sim y$, if either $x \prec y$ or $y \prec x$ in P; they are called *incomparable*, denoted $x \,||\, y$, if neither $x \prec y$ nor $y \prec x$ in P. If there is no z in X such that $x \prec z$ and $z \prec y$ then we write $x \prec\!: y$ and say x is covered by y or y covers x or (x, y) is a *cover* in P.

Every poset, strict or reflexive, has a dual resulting from flipping all pairs in the ordering relation. The dual of (X, P) is (X, P^d) with $P^d = \{\, (y, x) \mid (x, y) \in P \,\}$. Most of the concepts defined below come in pairs A/B (like least/greatest or infimum/supremum). They are related to the dual poset in the following way: the concept A is applied to a poset (X, P) is the same as the concept B applied to the dual poset (X, P^d).

A special case of a partial order is the *total order*. In addition to either of the three sets of properties above, total orders have the additional property that every element can be compared with every other element, that is, for every x, y in X, $x \neq y$, either $x \prec y$ or $y \prec x$. Total orders are also called linear orders because the elements of X can be arranged in a chain $x_1 \ldots x_i \ldots x_j \ldots x_n$ such that $x_i \prec x_j$ whenever $i < j$.

The number $n = |X|$ of elements in a poset is called its *order*. A poset of order n can have at most $\frac{1}{2} \cdot n \cdot (n-1)$ pairs in its order relation. This limit is reached for linearly ordered sets. True partial orders have smaller order relations.

Special elements. Given a poset (X, P), an element s in X is called

least	if $s \prec x$ for all $x \neq s$ in X	least *of all*, at most one
minimal	if there is no x with $x \prec s$	least *of those comparable*
greatest	if $x \prec s$ for all $x \neq s$ in X	greatest *of all*, at most one
maximal	if there is no x with $s \prec x$	greatest *of those comparable*

There can be any number of minimal and maximal elements. If a least (greatest) element exists, then it is also the only minimal (maximal) element of the poset. The set of all minimal elements of (X, P) is referred to as $\min(X, P)$ and the set of all maximal elements is referred to as $\max(X, P)$.

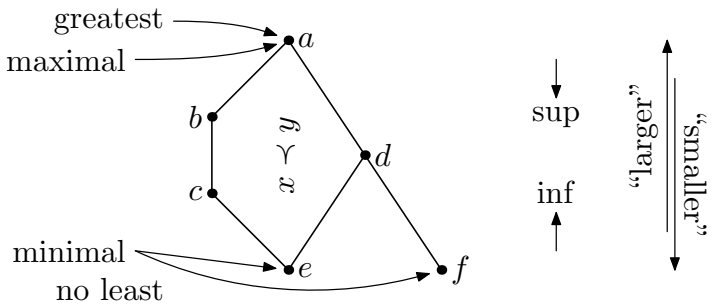


Figure A.1 Hasse diagram of a poset (X, P) with ground set $X = \{a, b, c, d, e, f\}$ and the ordering relation $P = \{\,(b, a), (c, a), (c, b), \dots\,\}$. The greatest element is marked. There are two minimal elements but no least element. This poset can be realised by the two linear extensions $e < c < b < f < d < a$ and $f < e < d < c < b < a$; $\dim(X, P) = 2$.

Subsets. Given a poset (X, P), any subset $S \subset X$ is again a poset under the ordering relation $P|_S$, which is the restriction of P to S (remove all pairs (x, y) from P for which x or y is not in S). If such a subset is linearly ordered, then it is called a *chain*. A subset of X consisting of elements that are mutually non-comparable is called an *antichain*.

We can now look at special elements relative to a subset $S \subseteq X$ of a poset. An element b in X is called

lower bound of S	if $b \prec s$ for all s in S (shorthand notation: $b \prec S$)
infimum	if b is the greatest lower bound
upper bound of S	if $s \prec b$ for all s in S (shorthand notation: $S \prec b$)
supremum	if b is the least upper bound

Infima and suprema are unique, if they exist at all. If $S = \{x, y\}$, then the infimum of S is also written $x \vee y$ and the supremum is written $x \wedge y$.

Sometimes, we are also interested in the set of all lower bounds (or upper bounds) of a given element e or a given subset S of the poset:

lower bound	$\downarrow e = \{\, x \in X \mid x \prec e \,\}$	$\downarrow S = \{\, x \in X \mid x \prec s, s \in S \,\}$
upper bound	$\uparrow e = \{\, x \in X \mid e \prec x \,\}$	$\uparrow S = \{\, x \in X \mid s \prec x, s \in S \,\}$

Finally, an *interval* $[p, q] = \{\, x \mid p \prec x \prec q \,\}$ consists of all elements "between" p and q. For a reflexive poset, p and q are included. Note that $[p, q] = \uparrow p \cap \downarrow q$.

Visualisation. Partially ordered sets can be visualised using *Hasse diagrams* (also known as *poset diagrams*, see Figure A.1). The elements of a poset (X, P) are represented by dots and if $x \prec y$ then the dot for x is lower in the diagram than the dot for y. Dots for those x and y form a cover in P are connected by a line. All other pairs in P correspond to strictly upward paths in the diagram. Technically, they can be reclaimed by computing the transitive closure (for reflexive posets also the reflexive closure).

A Hasse diagram can be made into a *directed acyclic graph* or *dag* by turning each line of the diagram into an edge of the graph, with the lower dot as the source node and the higher dot as the target node. This is also called the *transitive reduction graph*. The transitive closure of such a graph (add an edge for each path in the graph) corresponds to the original poset.

The reason posets are visualised "bottom-up" is that terms like "maximal element" or "least element" or "supremum" have an intuitive graphical representation. Unfortunately, this way of drawing a poset conflicts with the familiar way of drawing family trees (specifically lineal charts, not pedigrees), where $x \prec y$ would be read as "x precedes y" or "y succeeds x."

Linear extension. The elements of any poset (X, P) can be arranged in a total (or linear) order L such that whenever $x \prec y$ in P then $x < y$ in L. This process is called *topological sorting* and the resulting totally ordered set (X, L) is called a *linear extension* of the poset (X, P). For a real-world application of this process, think of a set of tasks that you have to solve alone. Some tasks depend on other tasks, so they are partially ordered. But as you can only solve one task at a time, you need the tasks linearly ordered.

Given a poset (X, P) and a linear extension L of P we write "$x \prec y$" if $(x, y) \in P$ (as before) and "$x < y$" if $(x, y) \in L$. These implications hold between all elements x and y in X:

$$x \prec y \quad \text{implies} \quad x < y \tag{1}$$

$$x < y \quad \text{implies} \quad x \prec y \text{ or } x \parallel y \tag{2}$$

Implication (1) is from the definition of linear extension and implication (2) is a warning that $x < y$ tells us only that $y \prec x$ is *not* in the poset!

Computing a linear extension for a poset (X, P) is simple:

1. choose any m from $\min(X, P)$
2. output m and remove it from (X, P)
3. repeat with step 1 until (X, P) is empty

This algorithm can compute a linear extension of a poset with N elements in time $O(N)$, assuming efficient access to the minimal elements at each

iteration. Actual implementations (like the one in [62]) use time $O(NE)$ where E is a function of the size of the relation, typically the number of edges in the transitive reduction graph.

How many linear extensions are there for a given poset (X, P)? Let $L(P)$ be the number of linear extensions for the ordering relation P. Then we have [112]

$$L(P) = \sum_{m \in \min(X,P)} L(P \setminus \{m\})$$

This formula counts the number of linear extensions by recursively removing one of the minimal elements. If (X, P) is the poset in Figure A.1 and $L^{xy\cdots}$ is a shorthand notation for $L(P \setminus \{x, y, \ldots\})$, then

$$\begin{aligned}
L(P) &= L^e + L^f \\
&= \left(L^{ec} + L^{ef}\right) + L^{fe} \\
&= \left(L^{ecb} + L^{ecf}\right) + \left(L^{efc} + L^{efd}\right) + \left(L^{fec} + L^{fed}\right) \\
&= L^{ecbf} + \left(L^{ecfb} + L^{ecfd}\right) + \left(L^{efcb} + L^{efcd}\right) + L^{efdc} + \left(L^{fecb} + L^{fecd}\right) + L^{fedc} \\
&= L^{ecbfd} + L^{ecfbd} + L^{ecfdb} + L^{efcbd} + L^{efcdb} + L^{efdcb} + L^{fecbd} + L^{fecdb} + L^{fedcb} \\
&= L^{ecbfda} + L^{ecfbda} + L^{ecfdba} + L^{efcbda} + L^{efcdba} + L^{efdcba} + L^{fecbda} \\
&\qquad + L^{fecdba} + L^{fedcba} \\
&= 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 9
\end{aligned}$$

Note that the superscripts show how all possible linear extensions are built.

The empty poset is vacuously linearly ordered and thus has exactly one linear extension, the empty set. A poset that is a chain has just one linear extension, namely itself. A poset that is an antichain of n elements has $n!$ linear extensions, namely every possible arrangement of the n elements. Every pair (x, y) in the ordering relation restricts the number of possible linear extensions because there can be no linear extension in which y precedes x.

Dimension of a poset. The intersection of all linear extensions of an ordering relation P equals P. Frequently, P equals the intersection of only a few (but well-chosen) linear extensions of P. (Remember that an ordering relation is a set of pairs; the intersection of two orderings consists of those pairs that are in both orderings.) A set of linear extensions of an ordering P such that their intersection equals P is called a *realiser* of P. Given a poset (X, P), the minimal number of linear extensions whose intersection is the ordering relation P, is called the poset's dimension, denoted $\dim(X, P)$ [26].

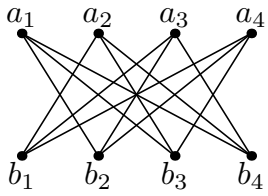


Figure A.2 The poset S_4 is one of a class of posets S_n known as the "standard example." It consists of n elements $a_1, a_2, \ldots a_n$ and n elements $b_1, b_2, \ldots b_n$ and an ordering relation that contains all pairs (a_i, b_j) for which $i \neq j$. The standard example has dimension $\dim S_n = n$. The poset S_4 can be realised by $a_1a_2a_3b_4b_1b_2b_3a_4$, $a_1a_2b_3a_4b_1b_2a_3b_4$, $a_1b_2a_3a_4b_1a_2b_3b_4$, and $b_1a_2a_3a_4a_1b_2b_3b_4$. Using the same scheme, a realiser can be found for any S_n.

The three linear extensions

$$L_1 = e < c < b < f < d < a$$
$$L_2 = e < f < d < c < b < a$$
$$L_3 = f < e < c < b < d < a$$

form a realiser for the poset in Figure A.1, but

$$L_1 = e < c < b < f < d < a$$
$$L_2 = f < e < d < c < b < a$$

is also a realiser and it is one with the fewest possible linear extensions. Therefore, the poset's dimension is $\dim(X, P) = 2$.

If $\{L_1, L_2\}$ is a realiser for a poset (X, P), then $x \prec y$ in P iff $x <_{L_1} y$ and $x <_{L_2} y$. Given a minimal realiser we can therefore test if $x \prec y$ in P in time $O(\dim(X, P))$. Another simple task is looking for least and greatest elements: an element is least (greatest) in (X, P) iff it is least (greatest) in all linear extensions of the realiser. Finally, if (X, P) is a poset with $\dim(X, P) = d$ and $S \subseteq X$, then $\dim(S, P|_S) \leq d$, that is, a poset's dimension cannot be increased by removing elements [126].These properties make realisers interesting for representing posets on computers, a topic to be investigated in Appendix B.

Unfortunately, there exist posets with few elements but high dimension. The upper bound for a poset (X, P) with $|X| = n$ elements is given by $\dim(X, P) \leq \lceil \frac{n}{2} \rceil$. The so called "standard example" S_n is a worst case: it has $2n$ elements and $\dim S_n = n$ (Figure A.2).

Lattices. A non-empty poset in which any two elements have a least upper bound and a greatest lower bound is called a *lattice*. In lattices, least upper bound is a binary operation on the elements of the lattice and usually called *join*. Join and precedence are related by the equivalence $x \preceq y$ iff $x = x \vee y$. Similarly, greatest lower bound is also a binary operation on the elements and usually called *meet*. Meet and precedence are related by the equivalence $x \preceq y$ iff $y = x \wedge y$. Finite lattices always have a least and a greatest element, which is a consequence of the existence of join and meet.

An *upper semilattice* is a lattice with only joins, that is, a poset in which any two elements have a least upper bound. Similarly, a *lower semilattice* is a lattice with only meets. A nice example from architecture is Alexander's comparison of the structure of historically grown cities versus artificially designed cities [1]. He found that designed cities have a tree structure (no overlap of city "units"), whereas historically grown cities have an upper semilattice structure, that is, city "units" may overlap and if they do, their intersection is also an element in the structure.

Summary of symbols. The following notation is used in this thesis:

(X, P)	a partially ordered set with ground set X and ordering relation P
$x \prec y$	x, y in X and (x, y) in P, "x precedes y"
$x \preceq y$	$x \prec y$ or $x = y$
$x \sim y$	comparable: $x \prec y$ or $y \prec x$ in the poset
$x \mid\mid y$	incomparable: neither $x \prec y$ nor $y \prec x$
$\downarrow x$	lower bound: the set $\{ z \mid z \prec x \}$
$\uparrow x$	upper bound: the set $\{ z \mid x \prec z \}$
$[x, y]$	$\uparrow x \cap \downarrow y$, the interval between x and y

Literature. Information about partial ordering can be found in most text books about theoretical computer science, [13] for example, and in books about discrete mathematics, such as [44]. Further information about linear extensions, realisers, and dimension can be found in [126] and [112]. My poset-related notation follows mostly the one used in [126].

In GIScience, partial orders are not widely used. There was some work done in the late 1980ies by Wolfgang Kainz [56,57] and the only textbook to my knowledge that (quickly) mentions partial orders is [136].

Appendix B

Computing a Realiser

Total orders are easy to represent on computers, because total ordering nicely corresponds to basic data structures such as arrays, lists, and—ultimately—the linear addressing of computer memory.

Partial orders do not have such a correspondence and consequently are hard to represent efficiently. Nevertheless, partial orders appear frequently in practice, both within computer science (for example, class hierarchies with multiple inheritance) and in various application domains. For example, the dependence relation among chapters in a book, the ordering of tasks in manufacturing automobiles, containment hierarchies, and even ontologies [55].

There are different approaches to representing posets. Probably most obvious is the use of a directed acyclic graph to represent the transitive reduction graph of a poset. However, with this approach even the simple precedence test operation $x \prec y$ is quite costly because it involves finding a path from x to y in the graph. Another approach is using bit-vectors $\varphi(x)$ to encode the elements x of a poset such that precedence in the poset corresponds to the subset relation among the bit-vectors: $x \prec y$ iff $\varphi(x) \subset \varphi(y)$.

Finally, we can use realisers. Realisers are appealing because they efficiently support many typical poset operations, including precedence testing and least upper bound computation. However, there is no known algorithm for computing a realiser given a poset.

This appendix develops a method to compute a realiser for a poset. The method needs as input an initial linear extension and a way to test if one element precedes another element in the poset. The resulting realiser is, in general, not a minimal realiser and there is, so far, no estimation how good the resulting realiser is. However, for the examples in this thesis the method proved to be well-suited.

Preliminaries. This appendix builds on the concepts and notations introduced in Appendix B. In particular, recall that a partially ordered set (X, P) consists of an irreflexive, asymmetric, and transitive relation P over a ground set X, and that a linear extension (X, L) of a poset (X, P) is a linear arrangement $x_1 x_2 \ldots x_N$ of the elements of X such that whenever $x_i \prec x_j$ in P we have $x < y$ in L.

In addition to Appendix B, the following notation is handy: if (X, L) is a linear extension, then it makes sense to talk of the i^{th} element, and we conveniently denote this as $L[i]$.

Algorithm. In order to compute a realiser $R = \{L_0, L_1, \ldots L_{K-1}\}$ for a poset (X, P) with N elements we require as input an initial linear extension (X, L_0) of the poset:

$$L_0 = x_1 < \cdots < x_i < \cdots < x_j < \cdots < x_N$$

Appendix A presented an efficient algorithm to compute a linear extension, so this requirement is no real impediment. We further assume there is a way to test for any two elements x, y in X if $x \prec y$ in P.

Stating that (X, L_0) is a linear extension for (X, P) means that $x_i < x_j$ in L_0 implies $x_i \prec x_j$ or $x_i \mid\mid x_j$ in P. The basic idea for the algorithm is to test each pair $x_i < x_j$ in L_0 for incomparability in P. If indeed $x_i \mid\mid x_j$ in P, we call this a *conflict* and *resolve* it by making sure that the dual pair $x_j < x_i$ appears in some other linear extension L_k with $k > 0$, creating new linear extensions if necessary. The tricky part is to not inadvertantly add $y < x$ to some L_k when actually $x \prec y$ in P. This problem is dealt with as follows. For a fixed i, define the two sub-orders of L_0: $inc(i)$ of elements incomparable with x_i, and $suc(i)$ of elements that succeed x_i in the poset:

$$inc(i) = (C_i, L_0|_{C_i}) \text{ where } C_i = \{\, x_j \in X : i < j \text{ and } x_i \mid\mid x_j \text{ in } P \,\}$$
$$suc(i) = (S_i, L_0|_{S_i}) \text{ where } S_i = \{\, x_j \in X : i < j \text{ and } x_i \prec x_j \text{ in } P \,\}$$

Based on $inc(i)$ and $suc(i)$ we define the total order $lin(i)$ as:

$$lin(i) = inc(i) < x_i < suc(i)$$

This order guarantees by construction that for any two elements x, y in $lin(i)$

1. $x \mid\mid y$ in P iff $y < x$ in $lin(i)$ and $x < y$ in L_0
2. $x \prec y$ in P iff $x < y$ in both $lin(i)$ and L_0

The second property ensures that $lin(i)$ is a linear extension for the original poset restricted to those elements in $lin(i)$.

This is not immediately obvious but can be proved as follows: Suppose $x_i \prec x_{j'}$ in P and $x_i \mid\mid x_j$ in P with $i < j' < j$, then $x_j \in inc(i)$ and

$x_{j'} \in suc(i)$. This reverses the order of x_j and $x_{j'}$ in $lin(i)$ with respect to the order in L_0, indicating that $x_j \parallel x_{j'}$ in P, which was not checked in the construction of $lin(i)$. However, if $x_{j'} \prec x_j$ in P, then by transitivity of posets, we also have $x_i \prec x_j$, contradicting the assumption that $x_i \parallel x_j$ in P. This is the only occasion in which pairs from L_0 are reversed in $lin(i)$ without being explicitly checked for incomparability.

Example. If $L_0 = x_1 \ldots x_i y z$ and $x_i \prec y$ in P and $x_i \parallel z$ in P, then $lin(i) = z x_i y$, switching not only (x_i, z), but also (y, z), which was not tested in the procedure described above but is fine as was just proved. □

The job of the proposed algorithm for constructing a realiser is essentially to embed $lin(i)$ for each i from 1 to $N-1$ into an existing linear extension, or, if this is impossible, into a new linear extension.

Before we can continue we must know how to represent linear extensions. A new linear extension L_k is created by allocating an array of N slots of memory and initialising the first $i-1$ slots with the first $i-1$ elements from L_0, for any conflicts involving those elements have already been resolved in the existing linear extensions. The slots i to N remain empty. A linear extension that still has empty slots is called *growing*, otherwise it is called *complete*. We define and use the following operations on growing linear extensions:

$add(k, x)$: if $x \notin L_k$ put x into the first empty slot in L_k, else do nothing;
$swap(k, i)$: exchange $L_k[i]$ and $L_k[i+1]$;
$pos(k, x)$: return the position of x in L_k.

The *swap* operation changes a pair to its dual pair, while leaving all other pairs unchanged; we also refer to it as an *adjacent transposition*.

Instead of explicitly computing all $lin(i)$ and embedding them, we can perform both operations on-the-fly while stepping through L_0 to look for pairs $x_i < x_j$ that are incomparable in P. This results in in the following algorithm:

```
1 K := 1   //number of linear extensions so far, including L_0
2 for i := 1 to N − 1 do k := 1
3   for j := i + 1 to N do
4     if ⟨x_i ≺ x_j in P⟩ then ⟨good, nothing to do⟩
5     else if k = K then ⟨create new linear extension⟩
6           if ⟨try ensure x_j < x_i in L_k⟩ then ⟨good, conflict resolved⟩
7           else k := k + 1   //try next linear extension
8                j := i + 1   //but redo all comparisons with x_i
```

```
 9    ⟨ensure x_i is in all L_k created so far⟩
10  ⟨ensure x_N is in all L_k⟩
```

Creating a new linear extension means to allocate a new array of N slots and initialise the first $i-1$ slots with the corresponding values from L_0:

```
    ⟨create new linear extension⟩≡
11    L_k := alloc(N)   //allocate memory for L_k
12    for l := 1 to i − 1 do L_k[l] := L_0[l]
13    K := K + 1   //one more linear extension
```

In order to ensure that $x_j < x_i$ in L_k we have to cope with several cases:

```
    ⟨try ensure x_j < x_i in L_k⟩≡
14    if ⟨x_i not in L_k⟩ then
15      add(k, x_j)
16      true
17    else //x_i already in L_k
18      if ⟨x_j not in L_k⟩ then
19        if ⟨x_i last element in L_k⟩ then
20          add(k, x_j)
21          swap(k, pos(k, x_i))
22          true
23      else //both x_i and x_j already in L_k
24        if pos(k, x_j) < pos(k, x_i) then true
25        if pos(k, x_i) + 1 = pos(k, x_j) then
26          swap(k, pos(k, x_i))
27          true
28    false   //cannot ensure x_j < x_i in L_k
```

If x_i is not yet in L_k we simply add x_j to L_k, knowing that x_i will be added later on by line 9. However, if x_i is already present in L_k, only three ways remain to resolve the conflict: (1) x_j is also present in L_k and precedes x_i, then the conflict is already resolved; (2) x_j is present in L_k and immediately follows x_i, then we can swap the two; (3) x_i is the last element so far in L_k, then we add x_j and swap the two. If none of these possibilities works out, then it is not possible to embed $lin(i)$ into L_k and we have to try another linear extension (lines 7 and 8).

While the loop on lines 3 to 8 adds all elements from $inc(i)$ to the linear extensions, lines 9 and 10 make sure that we also add x_i and the elements from $suc(i)$ so that in the end we are guaranteed to have the whole order $lin(i)$ embedded in one of the L_k.

⟨ensure x_i is in all L_k created so far⟩≡

29 **for** $k := 1$ **to** $K - 1$ **do** $add(k, x_i)$

⟨ensure x_N is in all L_k⟩≡

30 **for** $k := 1$ **to** $K - 1$ **do** $add(k, x_N)$

Proof of correctness. We need to prove two things. First, for each incomparable pair $x \parallel y$ in P there is a linear extension L_i in which $x < y$ and there is a linear extension L_j in which $y < x$. One of the two pairs is always in L_0, the initial linear extension. We know that the other pair also exists, because all pairs in L_0 are tested for incomparability in P and, if found incomparable, are guaranteed to be added to some L_k with $k > 0$ because ⟨try ensure $x_j < x_i$ in L_k⟩ is invoked for increasing values of k until it succeeds.

Second, whenever a pair $x \prec y$ is in P, it must also be in *all* linear extensions (L_k) of the realiser. If some pair $x < y$ in L_0 actually is in P, then x will be added to all linear extensions (line 9 and line 12) *before* y will be added. The swapping of adjacent elements that may take place in ⟨try ensure $x_j < x_i$ in L_k⟩ cannot disturb this arrangement because it is only invoked on pairs of elements which are known to be incomparable in P. □

Analysis of complexity. Time complexity is composed of these constituents:

1. $O(N^2)$ pairs in L_0 that are tested for incomparability in ⟨$x_i \prec x_j$ in P⟩;
2. this test is part of the input and thus out of our control—we assume it takes time $T(N)$;
3. if a pair in L_0 is found to be in conflict with P, we invoke ⟨try ensure $x_j < x_i$ in L_k⟩, which has to look up the positions of the x_i and x_j in L_k, and this is done in time $O(N)$ while the operations *add* and *swap* can be done in constant time;
4. if ⟨try ensure $x_j < x_i$ in L_k⟩ fails, we have to redo the inner loop, which means we have, in the worst case, three nested for-loops over N elements.
5. if a confict cannot be resolved using the existing linear extensions, a new one is allocated and initialised in time $O(N)$.

In summary, we get a total time complexity of $O(N^3 \cdot \max(T(N), N))$, where $T(N)$ may be as good as a constant (if the comparability test is implemented using lookup in an adjacency matrix) and hardly worse than $O(N)$ (if the comparability test operates on a hash table of the pairs of the order relation), so it is safe to say that the algorithm runs in time $O(N^4)$.

Note that computing the dimension of a poset was found to be an NP-complete problem [137]. Consequently, there cannot be a polynomial method for computing a minimal realiser for a poset, since that would amount to computing its dimension.

Space complexity is $O(NK)$ because we need space to store the K linear extensions, each of N elements. This neglects the storage required for the poset, which is provided by the calling instance and depends on the specific implementation.

What about trees? Much as every directed acyclic graph induces a poset (Appendix A), every tree (as a special type of a directed acyclic graph) also induces a poset. Such tree-shaped posets are always 2-dimensional, that is, can be realised with only two linear extensions.

Proof: If T is a tree with subtrees $T_1, T_2, \ldots T_n$, then the sequence $(T, T_1, T_2, \ldots T_n)$ and the sequence $(T, T_n, T_{n-1}, \ldots T_1)$ are both linear extensions of the poset and their intersection contains precisely the pairs (T, T_1), (T, T_2),$\ldots (T, T_n)$, that is, the original tree-shaped poset. □

The algorithm for computing a realiser does not, in general, find a minimal realiser for tree-shaped posets. The proof above directly shows how to build a realiser algorithm specialised for trees. However, since there are no trees in my thesis, this was not implemented.

Using the realiser. Let $R = \{L_1, L_2, \ldots L_K\}$ be a realiser for a poset (X, P) with $N = |X|$ the order of the poset and $K \geq \dim(X, P)$ the size of the realiser.

Represent the realiser by storing for each element $x \in X$ the positions x_i of x in each of the linear extensions L_i of the realiser. This can be done using K parallel arrays (or a single array of K-tuples).

For any two elements x and y we have $x \prec y$ in P iff $x_i < y_i$ in L_i for all i from 1 to K. Therefore, precedence testing requires time $O(K)$.

An element is least in P iff it is least in all L_i of R. If there is no element that is least in all L_i, then there is no least element in P. This follows directly from the definition of least element (Appendix A). Greatest element by analogy.

The lower bound $\downarrow x$ of x consists of those elements that precede x in all L_i. They can be identified in time $O(NK)$ by running through all elements. An optimisation strategy can be found in the source code. Upper bounds by analogy.

The supremum of two elements x and y is the least element in the common upper bound of both x and y; if there is no least element in this common

bound, then the supremum does not exist. This can again be computed in time $O(NK)$. Infima by analogy. Implementation sketch: set $m = \mathit{unit}$ (an internal greatest element); run through all elements z and check if $x \prec z$ and $y \prec z$; if so, and $z \prec m$, then set $m := z$; but if $m \parallel z$, then stop with error "no supremum." When done, m is the supremum, or it is still *unit*, in which case x and y are maximal. If the poset is reflexive ($x \preceq x$ for all $x \in X$), then there are two easy cases: if $x \preceq y$ then $x \vee y = y$ and if $y \preceq x$ then $x \vee y = x$.

If an element is least in one or more L_i of R, then it is minimal in P. The converse does not hold! Hence it is easy to find *some* minimal elements, but not all. The same holds for maximal elements.

The cover relation $x \prec\!: y$ holds iff $\bigcap_i L_i[x, y] = \emptyset$, where $L_i[x, y]$ denotes the set $\{\, z \in X \mid x < z < y \text{ in } L_i \,\}$. This can be computed in time $O(NK)$. A necessary (but not sufficient) condition is that $x \prec\!: y$ if x immediately precedes y in at least one linear extension. This is an easy-to-implement speed up for some cases.

List of symbols used in this appendix.

X	ground set over which the poset is defined
N	number of elements in X
P	ordering relation to be realised
L_0	an initial linear extension for P: $L_0 = x_1 \dots x_i \dots x_j \dots x_n$
K	number of linear extensions in realiser
L_k	the k^{th} linear extension of the realiser
i, j	indices into L_0 (we always have $i < j$)
k	used to index a specific linear extension in the realiser

Appendix C

The Odeon Software

The Schematic Geometry model (Chapter 4) and the software agents (Chapter 5) along with a graphical interface were implemented using the Java programming language. The resulting program is named "Odeon" after the ancient Greek theater places: the Odeon is where people and scenes interact.

The files that make up Odeon are bundled into a single Java archive file, Odeon.jar. To launch Odeon type "`java -jar Odeon.jar`" at the command line. Some operating systems allow you to simply double-click the jar file. Note that Odeon requires a recent Java version.

Figure C.1 shows a screenshot of the running software with the Enge data set (Section 5.3) loaded. The elements of the Schematic Geometry model are shown as rounded rectangles and the partonomic structure is indicated with lines between those rectangles. The currently selected element is shown with a heavy outline. Elements are dynamically coloured to indicate scenes, origins, destinations, etc. What was called an "element" so far is called a "node" in Odeon (and in the remainder of this appendix). Similarly, pairs of the ordering relation are called "edges."

At the top of the window are buttons to load and save Odeon files (see below), to validate the currently displayed model, and to quit. The **Toolbox** button brings up a dialog box with many functions to explore the Schematic Geometry model. The **Wayfind** button brings up the dialog box that is used to simulate wayfinding. Enter the desired start and goal node, choose an agent type, set options, the number of simulation runs, and click "Run" to start the agent. Alternatively, click "Step" repeatedly and observe what the agent does, step-by-step. The agent's wayfinding protocol that is shown in the lower half of the dialog box is explained in Figure 5.6.

The standard wayfinder's knowledge consists of those nodes that are selected in the "Nodes" tab of the Toolbox dialog. If no nodes are selected, then the default knowledge from Figure 5.7 is used.

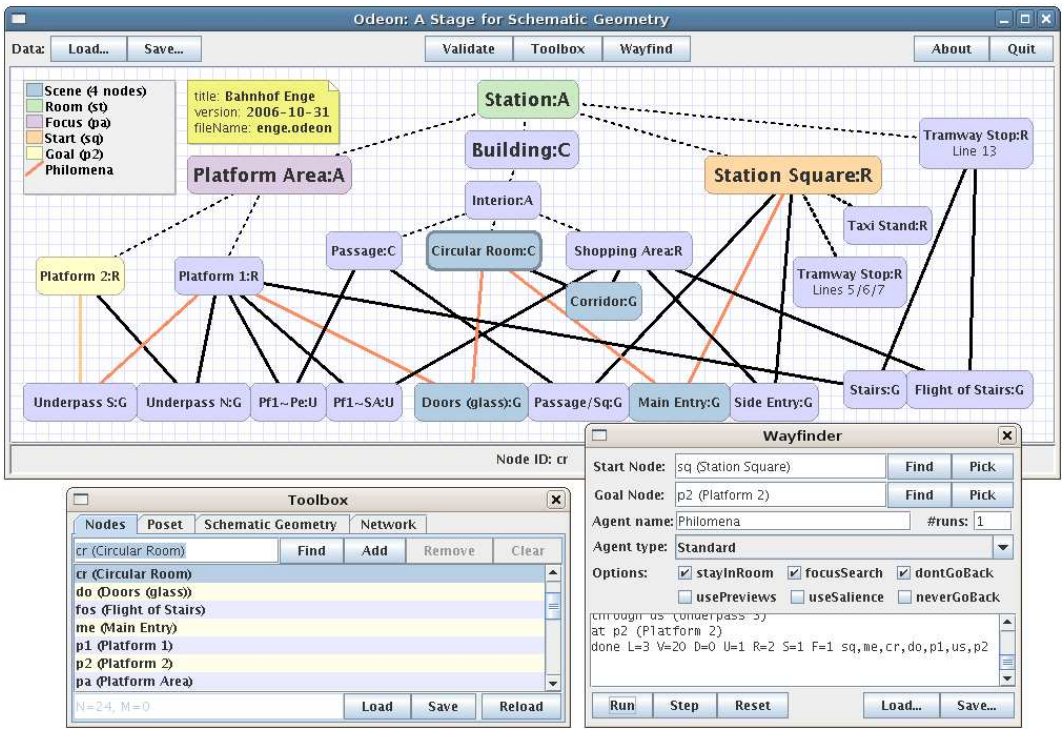


Figure C.1 User interface of the Odeon software

Schematic Geometry models can be created and modified with Odeon. Right-click on any screen element or into the background to open a popup menu with editing and configuration options. For example, you can add new elements, modify the partonomy, or change attributes of elements and connections between elements. Attribute changes are immediately effective. After structural changes you must click the **Validate** button to rebuild the internal indices.

The Odeon file format. Schematic Geometry models of an environment are stored in Odeon files. The Odeon file format is designed to be both human-readable and easy to parse by software tools. Odeon files are organised line by line. Each line consists of tokens separated by white space. The first token is a key that determines the line type and the interpretation of the remaining tokens. If the first token is a special character, then it need not be separated by white space from the next token. Empty lines are ignored. Unknown keys and invalid arguments for a given key constitute an error. These keys and line types are valid:

`#` *Bla blah...*	comment line, ignored
`+` *name value*	define attribute *name=value*
`node` *id schema* [*Name*]	node with given *id*, *schema*, and *Name*
`edge` id_A id_B	define the edge (id_A, id_B)

The *Name* in a node definition is optional; if omitted, the software will show the node's *id* instead of the name. Since tokens are separated by white space, they cannot contain blanks. If this is desired, the corresponding token has to be enclosed in double quotes.

Attributes belong to the node or edge whose definition most closely precedes the attribute definition. If no node nor edge precedes an attribute definition, then the attribute is considered to be global meta information and will be displayed by the user interface. A node's "geometry" attribute records its location on screen. The "salience" attribute is an integer number coding the salience of an element, typically a gateway; this attribute is attached to *edges* because the salience usually depends on the side from which a gateway is seen. The same holds for the "preview" attribute, which records the elements that can be seen looking through a particular gateway. Here are excerpts from the Odeon file for the Enge station (Section 5.3):

```
# Written by Odeon alpha (August 2006)
# Tue Oct 31 19:19:30 CET 2006
+title "Bahnhof Enge"
+version 2006-10-31
```

etc.

```
node sq REGION "Station Square"
+fontSize large
+geometry +744+102
node me GATEWAY "Main Entry"
+geometry +636+322
node pe CONTAINER Passage
+geometry +336+174
```

etc.

```
edge sq me
+salience 8
edge sq pq
+preview pe,pa
+salience 6
```

etc.

Running Odeon in batch mode. Instead of running simulations interactively, Odeon can be invoked with a single parameter that names a *command file* to be executed by Odeon. The command file consists of one command per line. The first character on each line defines the command and the remainder of the line is taken as the argument(s) to that command.

`a` *type*	make an agent of the given *type*: Standard, etc.
`c` [*option*]	clear *option* or all options if none specified
`f` *path*	redirect agent's output to *path*
`k` *list*	list of elements initially known by the agent
`n` *name*	define agent's name (purely informational)
`o` *option*	set the specified *option*: *stayInRoom*, etc.
`q`	quit Odeon
`r` [N]	run the simulation once (or N times)
`s` *element*	origin (agent's starting place)
`t` *element*	destination (the place to reach)
`w` *path*	the "world," path to a postore file
`z` *int*	sets the seed for the random number generator
`#`	comment line, ignored

The `r` command performs the simulation with the parameters that were specified *before*. Options, source, target, initial knowledge, and agent type are *not* reset after running the simulation; they have to be updated (or cleared) explicitly! Here is the beginning of the command file for the Enge simulations of Chapter 5:

```
# Batch simulating all Enge experiments
z 20070130

w enge.odeon
k st bg bgi pa sq

a Standard
o dontGoBack
o stayInRoom
o focusSearch

n 1.1
f t-1-1.wayf
s t567
t p2
r 1000
```

etc.

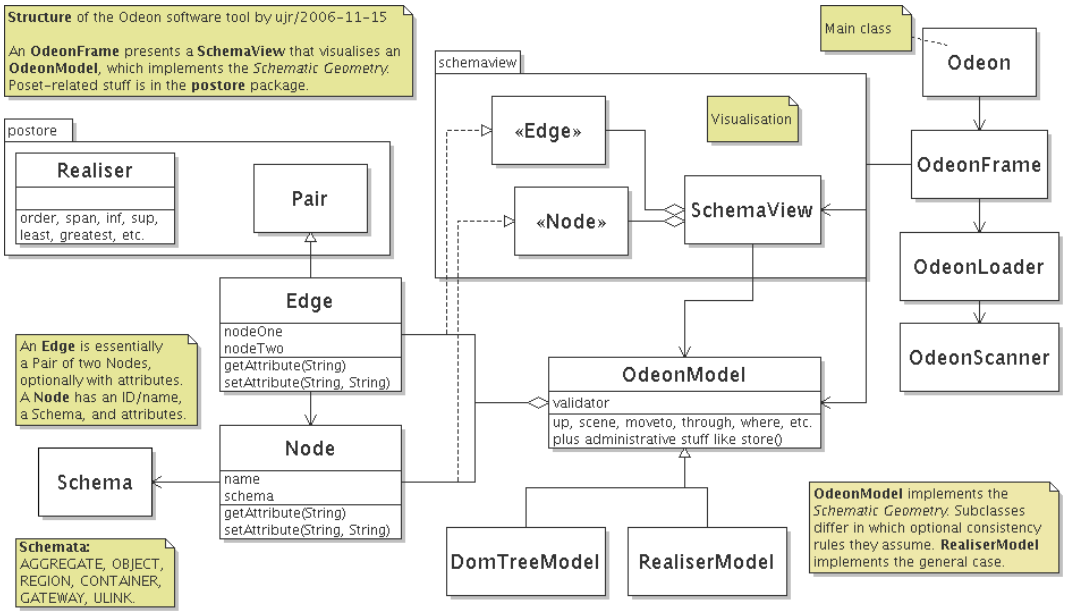


Figure C.2 UML class diagram for Odeon.

Implementation notes. Odeon is implemented as a collection of classes written in the Java programming language; Figure C.2 shows a UML diagram. As usual, most of the code deals with visualisation and maintenance of the user interface. Interesting source files are:

`Realiser.java` represents a realiser and implements poset operations.

`OdeonModel.java` contains the abstract class OdeonModel that implements those parts of Schematic Geometry that are independent of a particular poset representation, and

`RealiserModel.java` contains a subclass of OdeonModel that uses the Realiser class to fill in all representation-dependent details.

The remaining pieces of Schematic Geometry are provided by classes in the files `Schema.java`, `Node.java`, and `Edge.java`.

`StandardWayfinder.java` implements the standard wayfinder and contains the Java version of the chooseLink function (page 65).

The code makes use of generic types and enumerations. These modern Java features are only available in J2SE 5.0 and newer.

Appendix D

Glossary and Index

Fundamental terms of this thesis are collected here along with short definitions and pointers to the main text where they are introduced or used.

Cognitive Spatial Schemata: an abstraction of repeatedly occurring spatial elements like doors and rooms and regions; they build on image schemata but are specifically tailored to wayfinding in scene space; the primitives of Schematic Geometry (Section 4.1, p. 43)

Dominance: a place A is dominated by another place B if A belongs to B exclusively; everything located at A is then also located at B, which is known as "inheritance of location" (p. 51)

Image Schemata: mental patterns that help us making sense of our perceptions and actions by classifying and structuring them; introduced by Mark Johnson in 1987 [54] (Section 2.3, p. 23)

Knowledge: "ordered access to information" [81], which may come from several sources; in particular, there is the distinction between "knowledge in the head" and "knowledge in the world" [93]. Schematic Geometry can be used as a model of both, spatial knowledge in the head and in the world (Section 6.4, p. 90)

Locomotion: the physical aspect of navigation: maintaining direction, adjusting speed, avoiding obstacles, etc. (Section 2.2, p. 16)

Lynch, Kevin Andrew, 1918–1984: urban planner and architect, author of "The Image of the City" [77], where he coined the term "wayfinding" and reported the five elements of the city image (cognitive map of a city), nodes, paths, landmarks, edges, and districts

Means of transport: the specific vehicle used: *this* bus and *that* train, your car, my bicycle, etc.

Mode of transport: the broad class of vehicles used: train, bus, tramway, private car, going on foot, etc.

Navigation: purposeful movement, comprising locomotion and wayfinding (Section 2.2, p. 16)

Network space: an environment for wayfinding that exhibits a clear network structure, for example, an urban street system for car drivers (Chapter 3, p. 31)

Path: linear physical features in the world, along which travel is possible. Depending on the mode of transport, paths are given more specific names like streets or tracks. The existence of paths is typical of network space. The lack of paths is typical of scene space.

Preview: what can be seen beyond a gateway, the schematic counterpart of the geometric notion of a line of sight (p. 64 and discussion on p. 86)

Public transport: services for the transportation of people according to a predefined schedule, typically involving multiple modes of transport, and subject to published conditions of use (Section 2.1, p. 13)

Recognition: remembering an item when it is perceived, that is, recognising it; to be contrasted with *recall,* which is remembering an item that is absent [4]. In this work, it is assumed that the destination of travel, whether known or unknown, can always be recognised, though not necessarily recalled (Chapter 5, p. 58)

Robustness of wayfinding: the quality of being successful at wayfinding even with minimal knowledge and frequent interruptions (like a stimulating discussion on the way); humans are robust wayfinders because they rely on knowledge in the world and use qualitative representations, but the details are under-researched (p. 21 and discussion on p. 89)

Room Theory: a method to compute context-sensitive answers to where-queries (like "Zürich is a European city" to an American but "Zürich is in Switzerland" to a European citizen) devised in 1974 by cognitive scientist David E. Rumelhart. Source of the two main wayfinding heuristics used in this thesis (p. 62)

Route: a sequence of movement actions (**through** and **goto** for the agent developed in Chapter 5); in network space, this corresponds to a sequence of paths, whereas in scene space, a route is an alternating sequence of scenes and linking elements (gateways and unconscious links)

Satisficing: the phenomenon that humans usually are satisfied with a good solution, even if it is not an optimal solution when all factors are considered. Since humans lack the cognitive resources for global optimisation,

optimality is not possible. Term coined by Herbert Simon (1957: *Models of Man*, Wiley and Sons, New York; 1979: *Models of Thought*, Yale University Press, New Haven)

Scene: that part of the surrounding environment which can be seen and reached without conscious effort. The scene is thus defined in terms of visibility *and* accessibility. Within Schematic Geometry, the scene of a place is the smallest enclosing CONTAINER or the largest enclosing REGION if there is no enclosing container (p. 33 and p. 51)

Scene graph: a graph that expresses connectivity among scenes; it can be deduced from a Schematic Geometry model (p. 54)

Scene space: an environment for wayfinding that lacks a network structure but is dominated by nested open spaces, for example a station (with its hall, square, platform area, etc.), a shopping mall, or a public park (Chapter 3, p. 31)

Schemata: see Image Schemata and Cognitive Spatial Schemata.

Schematic Geometry: a model for representing the structure of scene space, using image schemata as its foundation (Chapter 4, p. 43)

Structure: how things are related to each other or composed from smaller parts; to be contrasted with *process*, which is about how things evolve over time; typical examples of structures are networks and hierarchies

Superimposition (of image schemata): the activation of more than one image schema for a given perception; introduced by [54] and analysed for the case of wayfinding in airports by [103]; some image schemata frequently occur together, for example, most PATH instances induce a LINK (p. 45)

Transfer (in public transport): changing the means of transport (and probably also the mode) because there is no direct connection from origin to destination; during transfers, travellers are always pedestrian navigators (Section 2.1, p. 13)

Wayfinding: the cognitive aspect of navigation, that is, the information processing necessary to travel from one place to another. Although a cognitive activity, wayfinding is always interaction with a (physical) environment (Section 2.2, p. 16)

Curriculum Vitæ

Name	Rüetschi, Urs-Jakob
Geboren am	5. September 1975
Heimatort	Suhr AG, Schweiz

Gymnasium 1988–1995

an der Kantonsschule Alpenquai in Luzern
Abschluss: Juni 1995 mit Matura Typus C

Studium 1996–2002

an der Universität Zürich seit Wintersemester 1996/97
Abschluss: Februar 2002 mit Diplom in Geographie
Diplomarbeit: "Denotative Geographic Modelling"
mit Auszeichnung der math.-nat. Fakultät

Doktorat 2002–2007

am Geographischen Institut der Universität Zürich,
als Doktorand angestellt: September 2002 bis Dezember 2006

Publikationen mit unmittelbarem Bezug zur Dissertation

U-J. Rüetschi & S. Timpf, 2004:
Schematic Geometry of Transport Spaces for Wayfinding,
Proc Münsteraner GI-Tage, *IfGI prints* **22**, 191–203.

U-J. Rüetschi & S. Timpf, 2005:
Modelling Wayfinding in Public Transport: Network Space and Scene Space,
Lecture Notes in Artificial Intelligence **3343**, 24–41.

U-J. Rüetschi & S. Timpf, 2005:
Using Image Schemata to Represent Meaningful Spatial Configurations,
Lecture Notes in Computer Science **3762**, 1047–1055.

U-J. Rüetschi & S. Timpf, 2006:
Cognitively Motivated Simulation of Transfers in Public Transport,
International Symposium on Transport Simulation, Lausanne, Switzerland.