
by

Carl Brown, Rune Gustavsson, Kristina Höök,
Per Lindevall and Annika Waern

July 1991

Swedish Institute of Computer Science
Box 1263, S-164 29 KISTA, SWEDEN
Final Report on
Interactive Route Guidance 1988-1991

Carl Brown, Rune Gustavsson, Kristina Höök,
Per Lindevall, Annika Wærn

SICS Knowledge Based Systems Laboratory for Prometheus Sweden - IRG
Box 1263, 164 28 Kista, Sweden
+46 8 752 15 00 (phone) +46 8 751 72 30 (fax)

T91:18

July, 1991
Executive Summary

In this report we present the more important research contributions made in the Interactive Route Guidance (IRG) project* carried out at SICS Knowledge Based Systems laboratory. The emphasis has been to look at those issues which affect the acceptability of the IRG system both from the driver's and society's point of view. These contributions include:

* a hierarchical representation of maps,
* a heuristic search algorithm for route-finding in a hierarchical space,
* a description of navigator stereotypes which may be implemented as user models in a navigational system,
* principles for description of routes to the resident-navigator,
* a methodology for the description of dynamic information that may affect traffic and route planning,
* an algorithm which tailors planned routes to constraints and considers dynamic information in the planning,
* a methodology for the presentation of route changes,
* a system architecture for the integration of the route planning mechanism with the mechanisms for planning and presenting routes suitable for human stereotypes, and
* a system architecture for the integration of in-car information systems.

The report is divided into chapters. The introduction gives a summary of the results reported in each of the chapters. Subsequent chapters are copies of the reports on specific aspects of the work.

Chapter 1  Introduction
Chapter 2  A System Architecture for Interactive Route Guidance
Chapter 3  Route Finding Using a Hierarchical Map Database
Chapter 4  Planning with Dynamic Parameters
Chapter 5  Directed Hierarchical Route Planning with Dynamic Information
Chapter 6  Presenting Route Guidance Information
Chapter 7  Results of on-Road Experiments
Chapter 8  Some Principles of Route Descriptions Derived from Human Advisors
Chapter 9  Human Routes - Routes for Humans
Chapter 10  A Conceptual Architecture for Dialogue Management
Chapter 11  Planning Route Guidance Dialogue

* This project was partially funded by Prometheus Sweden, a national programme of research and development in new technology for the automobile. The programme ran from July 1988 through June 1991. Prometheus Sweden was a part of the ongoing Prometheus initiative in Europe. Funding in Sweden was provided by STU, the Swedish National Technical Research Council, and the Swedish automobile industry, SAAB Automobile AB and VOLVO Automobile AB.

1 This paper is also published in the proceedings of Prometheus Workshop, Turin, 1990.
2 This paper is also published in the proceedings of Nässlingen workshop, 1990.
3 This paper is also published in the proceedings of ISATA-91.
4 This paper is also published in the proceedings of Prometheus Workshop, Turin, 1990.
5 This paper is also published in the proceedings of Nässlingen workshop, 1990.
6 A short version of this chapter has been published at the Cognitive Science conference 1991 in Chicago.
7 This paper is also published in the proceedings of Nässlingen workshop, 1990.
Final Report on Interactive Route Guidance
1988-1991

Carl Brown, Rune Gustavsson, Kristina Höök, Per Lindevall, Annika Wærn

SICS, Knowledge Based System Laboratory, Box 1263, 164 28 Kista, Sweden
+46 8 752 15 00 (phone) +46 8 751 72 30 (fax)
carl@sics.se, rune@sics.se, kia@sics.se, perl@sics.se, annika@sics.se

July, 1991

1. Introduction

In this report we present the more important research contributions made in the Interactive Route Guidance (IRG) project carried out at SICS Knowledge Based Systems laboratory. These contributions include:

* a hierarchical representation of maps,
* a heuristic search algorithm for route-finding in a hierarchical space,
* a description of navigator stereotypes which may be implemented as user models in a navigational system,
* principles for description of routes to the resident-navigator,
* a methodology for the description of dynamic information that may affect traffic and route planning,
* an algorithm which tailors planned routes to constraints and considers dynamic information in the planning,
* a methodology for the presentation of route changes,
* a system architecture for the integration of the route planning mechanism with the mechanisms for planning and presenting routes suitable for human stereotypes, and
* a system architecture for the integration of in-car information systems.

Much of this work is based on the development of cognitive models of routes and route planning. We believe that only on this basis will it be possible to develop products acceptable to the human driver. Many of the terms used will be explained in the ensuing sections of this report.

We begin by describing a scenario for how an IRG system might be used with one kind of user (section 1.1). Using the scenario, we outline the problems and research issues we have

---

1 The authors are listed by alfabethical order.

2 This project was partially funded by Prometheus Sweden, a national programme of research and development in new technology for the automobile. The programme ran from July 1988 through June 1991. Prometheus Sweden was a part of the ongoing Prometheus initiative in Europe. Funding in Sweden was provided by STU, the Swedish National Board for Technical Development, and the Swedish automobile industry, SAAB Automobile AB and VOLVO Automobile AB.
identified (section 1.2). We then summarize the results of our research (section 1.3). In chapters 2 through 10 we have included the detailed research reports of the project. In them, a thorough description of the research results may be found.

1.1 Scenario

“The taxi driver standing by a post at Medborgarplatsen gets a message from the receptionist to go to Sveavägen 30 to pick up an old man and his wheel-chair. At the same time, the IRG system picks up the message containing the destination. Using the destination and the recently received information about the traffic jam at around Klartunneln, the IRG system computes the best route. If there had not been a traffic jam, the system would have computed the route to be Söderleden, Klartunneln and up Sveavägen. Instead, the IRG system chooses the route to be Skeppsbron and then by small streets up to Sveavägen. The system switches on the Head-Up Display (HUD) in front of the taxi driver, and also switches on the sound system. It then puts up a textual message at the screen saying that the a good idea would be to: “take Skeppsbron and then up Birger Jarlsgatan”. The message is short, and leaves out a number of roads and intersections, assuming that the driver knows the city quite well since she is a taxi-driver.

The taxi-driver at this point wonders why would she go via Skeppsbron, instead of Söderleden and Klartunneln? At the bottom of the screen there is now a message saying that the route was chosen since there was a traffic jam around Klartunneln, which is avoided by the present plan. So, she accepts the suggested plan and gets going.

The taxi driver, who is from Dalarna, is a bit unsure about the route after Skeppsbron, and so when having passed Skeppsbron, she switches on the navigational help system. It helps her through every intersection by displaying which turn to make on the HUD and telling her on the loudspeakers, until she reaches the destination. On the way, the IRG system is notified by the traffic control center that one of the roads ahead is blocked by a lorry. A new plan with a short detour from the current plan is calculated, but, since the taxi-driver has chosen to switch on the navigational help system, the IRG system decides not to notify her that a new plan is now being executed. The initial description to go via Skeppsbron and Birger Jarlsgatan is not changed anyway.

At the destination, while the taxi-driver helps the man to get in and puts the wheel chair in the trunk, the system looks up where the man has told the receptionist he wants to go when he phoned in his request for a taxi. It computes the route, and displays a new message for the driver as soon as she is back in the car.”

1.2 Research Issues - Definition of the Problems

What are the system functions we can identify from this scenario? First and obviously, planning is involved; both strategic planning but also dynamic replanning in which the route is re-calculated during the trip. Second, we need to describe routes and actions to the driver, sometimes before the trip, sometimes during the trip and sometimes both before and during the trip. Third, the plan described needs to be adapted to the driver, perhaps in more detail when required. Finally, although partly invisible in this scenario, some reasoning is going on to determine that the drivers intentions are equal to the destinations received from the receptionist, about what to do when the system comes up with the new plan, and how much information that has to be transferred to the driver, etc.

How to achieve this functionality together with the issue of dialogue management (integrating several driver-information units in the car) have been the issues dealt with in this project.

The research directions we took up were the following. The emphasis has been to look at those issues which affect the acceptability of the IRG system both from the driver’s and society’s point of view. From that standpoint there are four broad areas defined below. In our work, we found strong relationships between these areas; any development in one area has had an impact on the others.
1.2.1 The research issues

Strategic and reactive planning:
- What are the characteristics of the strategic planning involved? How does the strategic planning in IRG differ from classical planning methods?
- How can we include environmental and societal constraints on the planning procedures, both in the strategic and the reactive planning?
- How can the planning be done in such a way that the driver can take effective guidance from the results of the planning?
- What is the nature of the reactive planning required? How is this reactivity different from existing reactive planning paradigms? Can we generalise the reactive planning to incorporate several kinds of dynamic information?
- Can we incorporate the reactivity required into the strategic planner? (Or is a separate mechanism needed?) Can reactive planning be localised to a small area near the current geographical position?

Conceptual models of routes and plans:
- How does a human driver conceptualize a route description? How does this conception vary between different types of drivers?
- Can we make strategic and reactive route plans that correspond to a human driver's conception of an acceptable (or better, a good) route?
- What are the principles for describing a conceptual route plan to a driver? Do such principles vary between driver types?
- What kind of map data structures will support such conceptualizations and planning?
- How do we describe a route to a driver before the trip is started? To which type of driver is this useful?

Reasoning about routes and changes:
- How do we reason about when to inform the driver of a change in a planned route? How does the reasoning depend on the type of driver?
- How can we reason about a driver's intentional changes in a route? When do we inform a driver that she is off the planned route?
- How do we reason about the correspondence between a driver's actions and the current plan?

Architecture for the dynamic replanning IRG system:
- What is an appropriate conceptual architecture for a knowledge-based computer system to support this kind of planning for human needs?
- What is the division of competence between the components of such a system?
- How can such a system be integrated with other information systems in the automobile?

1.2.2 Research approach

In our IRG-project at SICS we have decided to regard navigation as a two-agent system, as opposed to the traditional system-user view. One agent is the IRG itself. The other is the driver. With the division of the system into two agents we can treat the driver as an intelligent, independent actor, who sometimes will understand and perform the suggested guidance instructions, and sometimes will not understand an instruction or not want to follow an instruction for various reasons. We have placed an important restriction on the IRG's interactions with the driver. The IRG is only suggestive. This means that:
- It does not affect the traffic environment directly, and
- It only informs the driver. It does not impose a choice on the driver.

We must realize that even with the most advanced, adaptive system, which has all available data, the human driver will sometimes know more about the traffic situation at hand, and will therefore be a better judge of what to do, or will not accept to be the slave of such a sys-
tem. It is therefore necessary that the system has a more or less explicit model of the driver. We have realized the driver model in two ways, firstly as an explicit user model, and secondly through the way the system reacts to diversions from the route planned by the system.

What kind of information does the driver need to transfer to the IRG system? What kind of information is it that we want to transfer to the driver? The problem falls into two parts or phases: the trip planning phase and the route guidance phase.

The trip planning phase, is where the system plans a route that meet certain criteria. The criteria to be met can be several. Most important are, of course, that the starting point coincides with the automobiles location and that the drivers destination is reached through execution of the plan. The drivers other criteria may be specific. They can, for example: include time constraints for when the destination should be reached; require that the route is the most scenic, the fastest, the shortest, the least gas consuming, etc; or that there is a sequence of destinations (a tour) such as passing a gas station or a restaurant on the way.

Apart from the user-specified criteria, the authorities might wish to impose criteria on the routes chosen by the system (for instance, that they pass through areas which are less sensitive to pollution, or that major roads are used instead of small local roads whenever possible).

From a human-factors point of view, it might be important that the structure of the planned route fulfills certain criteria, such as not including too many left turns, keeping to major roads instead of minor ones, etc. Another criteria might be that the route can be easily described to the driver. What is meant by easily described is dependant upon who the driver is.

When all the criteria have been established, a route can be planned. We might then decide to present the entire plan to the driver before leaving, i.e. within the planning phase, or we might decide to split the plan into small portions, and give it to the driver one portion at a time during the second phase, the route guidance phase. What we want is to minimize the amount of work that the driver must undertake during the trip, since any extra workload might threaten safety. We have identified the need of different descriptions and also different routes choices for different kinds of drivers. As a first rough division we can differentiate between; drivers who are unfamiliar with a city, which will be called tourist navigators; drivers who live or work in a city, residents navigators; and drivers normally traveling back and forth between a few destinations, commuter navigators. Of course, there is no clear-cut line between these three groups. You might even belong to the first two groups in one city, for instance, if you know one part of the city very well, and another part not so well. Below, we shall see in what respect the presentations and route choices might be different for these three groups.

1.3 Solutions

The problem areas outlined above has been attacked in a number of working documents in our project. These documents form the basis for chapters 2 - 10. Here we give a brief outline of the solutions.

1.3.1 A Conceptual Architecture for the System

First, what provides a unified view of the research effort is an outline of a system architecture, described further in chapter 2, A Conceptual Architecture for Dynamic Route Planning. The division of work supported by the architecture has made it possible for us to isolate a number of research issues, and also to implement some solutions. A large part of the system architecture has been implemented in SICStus Prolog code and runs under UNIX on Sun workstations at SICS-KBS.

The architecture we propose is a system of concurrent processes, where the processes are defined from their respective competence. As a vehicle to viable solutions we have identified a typical scenarios (discussed later). The implementation is a set of concurrent processes. The set of processes are the same during all scenarios (as opposed to a more object-oriented approach, where processes could be started or terminated when needed).
In figure 1 we see a diagram of the architecture. The arrows represent data flowing from one process to another. The heavy arrows represent a triggering signal that causes the route-planner to replan.

![Diagram of the system architecture](image)

**Figure 1.** The system architecture.

The database contains all the necessary information for doing planning taking dynamic information into account, as well as user preferences and other constraints. It also contains the information needed to communicate efficiently with the driver, a user model. The dynamic monitor will look out for new dynamic information which influences the current plan, and if it does this in a significant way it will trigger the route planning unit. The route monitor will check that the plan is followed through checking the location of the car against the plan. It can also trigger the route-planning unit if the car goes off-route. The route-planner is responsible for computing the route plan. The HMI-planner presents the plan to the driver (through the car's HMI mgt. system), both before and during the trip. It plans the dialogue, and it adapts to the driver profile. The meta-planner controls the various units, for instance, checking to see whether a new plan is different enough from the current plan to be worth installing.

### 1.3.2 Planning - Algorithms and Knowledge Representation

In chapter 3, *Route Finding Using A Hierarchical Map Database*, we present the basic system of strategic route planning. We have developed a hierarchical map structure which mirrors a cognitive model of maps; higher levels give an overview of the map, succeeding lower levels give more detail. We as humans view a road map hierarchically. We see the map first as a system of route segments as linking major centers or intersections. This is often mirrored in the actual printing of the map, where major routes are indicated with a prominent colour and an easily identified symbol (such as a double line). Details of a map are often abstracted into separate city-centre maps or are found in smaller-scale maps. Humans plan routes and trips with these maps. We have developed a system which plans in the same way.

In the map structure, all levels of the representation are isomorphic; that is, at any level the same primitives may occur. The map primitives are nodes and segments. Nodes may be of two types; intersection nodes or expandable nodes. Segments are directed arcs. Associated with the expandable nodes are cost-of-traversal estimates for the entry-exit-point pairs. This enables us to plan at higher hierarchical levels without expanding the nodes.

The search algorithm presented in this chapter makes use of the hierarchical map to do more detailed planning when needed. The search algorithm which operates on the map structure is
of a best-first type which evaluates each route choice heuristically on its total path estimate. In order to verify our data representation and algorithm performance, a flat (non-hierarchical) version of our map was built that, together with an A* search algorithm, formed a basis for comparison with our own work. We found that our hierarchical approach out-performed the standard A* algorithm in almost all circumstances.

Chapter 4 on Planning with Dynamic Parameters studies the feasibility of dynamic constraints being represented as areas. A study of traffic flow parameters allowed us to represent dynamic traffic information as parameterized areas.

A notation for traffic flow parameters, introduced in [Gus89], is used to denote the influences of weather etc. Traffic flow parameters of length, d, expected travel time, Δt, traffic flow, F, density of vehicles, R, and capacity of the segment, C, are used to describe segments. These parameters are all variable with time and dynamic factors. We introduce the k-value parameter and show that the dependency on dynamic factors may be modelled by the k-value. Thus the set of parameters (dk, Δtk, Fk, Ck) is parameterized itself by the k-value.

The most striking property of the area representation is that its adaptation to the hierarchical map representation and the accompanying search algorithm is linear. That is, the dynamic search algorithm collapses to the static algorithm in the absence of dynamic information. Chapter 4 explains in detail the information needed to build area representations. Weather patterns that change with time, traffic congestion that varies hourly and daily, pollution that may build up and then disperse over a wider area and traffic incidents that have an effect on the surrounding traffic patterns are examples of the kind of dynamic information that may be represented as areas.

Chapter 5, Directed Hierarchical Route Planning with Dynamic Information, describes how the dynamic traffic areas are taken into account by the adapted search algorithm. The extra cost (in time for example) is encoded as impedance factors. Impedance factors and the overlap of areas with the planned or partially planned route are used to compute the total cost of a route choice. The directed search is governed by the impedance associated with areas. Planning may be divided into two phases; the initial plan and plan refinement.

We have taken the basic A* strategy and adapted it for our hierarchical map representation. This provides a level of initial plan generation. We find an initial path that moves from the starting node, at level N of the hierarchy, to a final node at the same level or higher. The initial path never drops down a level in the hierarchy. It may traverse the graph at the same level, or climb a level as nodes are added to the path.

The initial plan contains intersection nodes and expandable nodes. For plan refinement, the initial plan is investigated with respect to expandable nodes. If any expandable nodes are found, planning the path at the lower levels of the hierarchy is done. (We would like to point out that this may be done "when necessary", and need not be done immediately after planning at level N only. Thus, possibilities for lazy plan refinement exist and can be exploited on route.)

To cope with dynamic parameters we use the strategy, for finding optimal routes, which uses an impedance factor. Impedance is associated with dynamic areas that the path will traverse on its way to the destination. Areas of easy access have a low impedance while areas of high impedance have a higher impedance. A path that considers crossing an unfavourable area will receive a higher total cost than the same path would have received if the area was not considered. The evaluation function can be explained as a terrain navigation or obstacle avoidance procedure. The expression for the cost evaluation function that considers impedance of dynamic parameters is:

\[
\text{TOTAL\_COST} = \text{PAST} + \text{FUTURE}(1 + \text{IMPEDANCE})
\]

where IMPEDANCE is based on the amount of overlap, area content and the weight-factor for the area. Past and the future costs are evaluation is described next. The impedance is a factor based on the amount of overlap a straight line makes over the predicted area from the current planning position to the destination. The overlap to straight-line ratio weighted with the weight-factor for the area is the impedance for the area. In multiple weight-factor predictions the impedances is simply the summation of the part impedances. With several predic-
tions the impedance is the sum of the impedances of each prediction. Formally it is described by:

\[
\text{IMPEDEANCE} = e_i \left[ e_j \left( \frac{\text{overlap}_{i,j}/\text{straight line}}{k_{\text{default}}/k_{i,j}} \right) \right]
\]

where

1, j is area j of prediction i

\( \frac{\text{overlap}_{i,j}}{\text{straight line}} \leq 1 \)

\( e_j \left( \frac{\text{overlap}_{i,j}}{\text{straight line}} \right) \leq 1 \) at a fixed i

\( \frac{k_{\text{default}}}{k_{i,j}} \geq 1 \)

Dynamic replanning of routes is concerned with predictions and plan refinement. The interplay between the three can be complex. We have examined the possible effects. We believe, after having developed a conceptual architecture for a complete dynamic replanning navigation system, that the implementation of such a system requires careful attention to the needs of the driver. The following sections explore this.

1.3.3 Initial Studies into Human Navigation Needs

In chapter 6, Presenting Route Guidance Information [Wärn90a], an outline of the fundamentals tasks for a route guidance system is given. It introduces the work carried out in chapters 7, 8 and 9 and forms the basis for the division of drivers into the first two groups mentioned above, namely the tourist and the resident navigators.

For tourist drivers the important issue is to give help sufficient to get from the starting position to the destination during the trip. For resident drivers the goal will be to provide a better route choice than the one the expert would have chosen, or to aid the driver in a complex route choice situation. How information best should be presented, depending on what descriptions the driver can understand. It is claimed that novice drivers will perform better if given egocentric instructions and have little need of beforehand information, whereas expert drivers can make use of information given before the trip and have less need of time-critical instructions. In the available literature an assumption is usually made that the driver would be a tourist. We identified the need to look further into the needs of expert drivers.

The paper established a need for system knowledge of dynamic data and further investigation of the drivers needs in route guidance. Finally, it is concluded that a route guidance system must be able to reason about the driver's knowledge of routes. For this reason, it is necessary to construct a formal model that allows reasoning about the driver's knowledge, that is powerful enough to handle the on-route situation where the driver departs from the expected route.

Chapter 7, Novice Route Descriptions: Pre-Study Results [Wärn90b] reports a pre-study on tourist navigators (called novices in the report). The purpose of this study was to find out what questions must be further studied. Firstly, the study shows (as expected) that close-range, local directions work for aiding tourist navigators. The drivers were fairly content with a co-driver who give instructions so long as they were unfamiliar with the area. They did not demand explanations for the route choice, and very seldom requested location information. We also identified the confirmation phenomenon. This occurs when the driver needs to recheck that an instruction is what he understood it to be. One way of giving the driver access to confirmation is to have the instruction accessible continuously. It might for example be shown on a display as long as it is valid.

This pre-study indicates that finding a landmark to describe a known place might be a bigger problem than actually finding out what places people know of. One important aspect of a route guidance system is how it can help a driver to get acquainted with an area. In this pre-study, we noted that drivers at times tried to relate their previous knowledge to the current trip, when they came to places they recognized.
The pre-study has given very little indication as to how far in advance drivers need instructions. It is important to do field studies on this to establish safe and acceptable timing limits. It seems that very short forewarning times, in the range of 5 seconds, will do (depending on the speed of the vehicle). The prestudy indicates that there are two limits: one minimum-time limit which gives the last moment in which an instruction can be given, and one environment-dependent limit which gives the earliest situation in which the instruction can be given in order to be remembered and carried out correctly. Finally, the instruction contents is dependent on where the instruction is given. For example, in natural language, the same turn command can be worded alternatively as: "Turn right at the intersection after this one", "Turn right in the next intersection", "Turn right at the intersection", or "Turn right here". Similar "format" differences occur with graphical displays.

1.3.4 Describing Routes to Humans

In *Some Principles of Route Descriptions Derivved from Human Advisors* [Höök and Karlgren], chapter 8, an extensive study on residents describing routes in Stockholm was made. In it we exposed the underlying principles of citizen and tourist route description done by humans, both in terms of how they are described and, to some extent, why. We also showed how the actual route choice differs for the two groups.

Furthermore we indicated how these principles could be implemented and used as an interface to a route guidance system. We believe that the solution we outlined is good for citizens, but not so good for tourists, but this needs further inquiries.

In the study, three different kinds of route descriptions showed up. One declarative aimed at residents, another procedural aimed at tourists and finally a mixed description which is procedural in nature, but since it is aimed at residents it is very shorthanded.

The descriptions aimed at tourists basically consisted of the following elements:

- The basic unit for describing a route is an instruction to perform an action.
- Actions are choices made in an intersection.
- Instructions can also be placement instructions and maintain instructions.

There were some heuristic rules derived as to when any of the above rules comes into play. For instance, when describing an intersection, the name of intersection is most important, then comes type, landmark and size and thereafter intersecting road and number of intersections.

Let us try and summarize the declarative descriptions. What is interesting about declarative descriptions is that they are extremely short. The instruction is not adorned with much extra information, and is declarative rather than procedural. Roads are left out in the middle of the trip, the start of the trip, and the end of the trip. Those gaps can be expressed in some heuristic rules:

- The first 1 - 3 roads before a road on a higher hierarchical level can be taken away from the declarative description.
- If the goal is on a high hierarchical level, the last 1 - 3 roads can be skipped.
- In the middle of the trip, only roads that help excluding other alternative routes are mentioned, plus some roads that help making the route description ‘complete’ (these road are usually on a high hierarchical level and it would be odd not to mention them).

There are also some principles for when a switch to mixed descriptions is made. It occurs when a ‘strange’ route has been chosen. Something seems to be a strange route whenever:

- roads on a low hierarchical level are chosen,
- or when the chosen route is much longer than the “as the crow flies” distance.

Finally, the citizen pattern: “drive Road1, drive Road2 ...”, is only changed using a landmark or hierarchical information on a few occasions. The principles appear to be that:

---

3 Some existing navigation systems (for example CARMINAT) use the speed limit as a factor when deciding the minimum forewarning time. Evaluating studies of these systems is perhaps the most valuable way to gaining further knowledge about this.
- A road name can be replaced by a landmark that uniquely determines that road when the subject can not remember the name of the road, or when the landmark is such a known item that its name would supersede the road name.

- Whenever the hierarchical level of the road is unusually low, it indicates that a strange road has been chosen, and to emphasise this the hierarchical level of the road is mentioned.

Route choices were sometimes different for tourists as opposed to residents, the underlying principles seems to be that:

- Tourist routes are chosen by subjects in order to be easy to explain. A route is easy to explain when it is on a high hierarchical level, contains few turns, and goes by important landmarks.

- Routes are constructed by tracing the route backwards from the goal to a point that is easy to connect with the starting point.

1.3.5 Planning Routes for Humans

In chapter 9, Human Routes - Routes for Humans, a solution to how to make human-adapted route planning in Interactive Route Guidance (IRG) systems is outlined. Firstly, different driver needs are motivated and summarized. A solution using the hierarchical map representation is then presented.

In available Route Guidance (RG) systems, the way the route is planned, is usually to find the shortest route between two points. The shortest route will be the shortest in terms of time or distance. We could claim that if we had a perfect RG system, the driver should simply sit back and follow the directions given by the system, and she will get to the destination on the quickest or shortest route without having to plan or understand the route herself. Unfortunately, the shortest or quickest route might be horrible to drive. It might contain many left-turns, it might go on small, narrow roads, etc. The driver might also be unwilling to simply follow the directions. Instead she will abandon a route suggested by the system if it diverts too much from her own opinion of the best route choice.

Therefore, the total sum of costs for a trip used by the planning-part of the RG system, must reflect the desires of the driver, apart from reflecting the desires of the society. In addition to the drivers needs above, the society might advocate routes that minimize pollution and/or accidents and so on. It should be noted that the introduction of advanced navigation systems along the lines we are discussing can drastically change the behaviour of both the drivers and the society. The effects of such systems has to be considered and estimated along with their design and introduction.

The behaviour of present day drivers performing navigation tasks has been studied by several researchers. Elliott & Lesk [Ell82] report that drivers strategies typically include finding main roads and that they stay on these as long as possible, that is, as long as the general direction towards the destination prevail. They also found that humans dislike computer planned minimum-distance routes because they contain far to many turns. Streeter & Vitello [Str86] report that the two most important factors to minimize on short trips were road construction/bad roads and traffic. For longer trips the important factors were road construction/bad roads, use of main roads, total distance, good scenery, few stop lights and travel time.

From these papers one can conclude that drivers rate factors that promote comfort and ease of driving higher than total distance and travel time when planning a trip. Some of these factors are static and imbedded in the hierarchical map database and to achieve routes that reflect the above mentioned preferences is a matter of search algorithm design. The truly dynamic factors are traffic density, mean speed and travel time. To be able to include them in the cost function they have to be expressed explicitly for each segment of a route. For the dynamic parameters, such as weather, our area representation achieve the same means.

From the society's point of view, traditional means to achieve traffic control is through directing the traffic to certain roads/streets and limiting the access to others. These means are inherited by the map database and are static. With the area representation a new mean for dynamic traffic control is at hand. As the situation is today, certain city regions have traffic in-
formation broadcast from a local radio station. To be able to use the information broadcasted, the driver have to stay tuned to a specific radio station, he have to understand what is said and he must have reasonable good local knowledge to be able to reason about the effects of the message. This implies that there is a great chance that local traffic messages are not of any help for visiting drivers. The area representation can express the desires of the society in a way which is transparent to the driver and which is unaffected by his choice of radio station.

In this chapter we examine some of the criteria used by humans today when planning routes, some studies by others, and one study by us. We also look at the few reports that have investigated how humans react to computer-planned routes done on existing RG systems. We then outline a solution as to how both the society and human criteria of route planning can be taken into account by a planner based on a hierarchically organised database.

1.3.6 Integration of Information Systems: Dialogue Management

The integration of an information systems, such as IRG, into the automobile is considered in Dialogue Management and Integrational Priorities, Chapter 10. In our work on the dynamic re-planning of routes we have become aware of the need to integrate the dynamic-replanning functionality with the rest of the navigation system in a way that responds naturally to the drivers requirements for navigational help. In addition, we have become aware of the absolute necessity of integrating the navigation system with all other information systems in the car.

In this investigation the following themes were developed. A general concept of Human Machine Interface (HMI) management was motivated on the basis of safety and extendability of information systems. Message parameterization for all information was developed. A conceptual architecture for dialogue management process was developed to ensure prioritization of messages and timely delivery of messages. Finally a standardized message semantic is suggested as a means of dealing with extendability and message presentation.

For HMI management there are two motivations. The first is safety. The second is engineering. In-car information systems are becoming increasingly complex and increasingly numerous. All this information is competing for the driver's attention with the situation outside the car and with the driving task. For engineering and cost-effectiveness consideration, modular design and extendability of systems are essential. This is especially true in the fields of systems development and software engineering, which are the fields involved in the development of dynamic replanning IRG systems and of most Prometheus driver information functions.

A dialogue message is the term we use for any information passing between the driver and vehicle systems in either direction, in any format. Messages:

- move between a diversity of systems and the driver.
- originate from a diversity of controls, systems and interface devices.
- may be presented in a variety of ways to the driver.
- contribute to the driver-workload.
- must be ordered in time and in space.
- have priorities.

The schema in figure 2 below gives a conceptual view of the HMI-Mgt process.

Messages must be parameterized in order to ensure extendability and a uniform treatment of priorities. The parameterization of messages we arrived at is discussed in some detail in the chapter. Required parameters are: Destination-process, Source-process, Type, Priority, Timestamp, Workload_Factor, Workload_Weighting, Duration, Persistence, Expiry, Medium, Contents and Opacity.

The Dialogue Management process must manages a set of messages to be transmitted; the Pending_Msg_Set. In addition, the process must be aware of the current set of messages that have been transmitted and whose lifetimes have not expired: The Current_Msg_Set. New messages may arrive to swell the Pending_Msg_Set. Messages may be transmitted which reduce the Pending_Message_Set and swell both the Current_Msg_Set. Messages may expire due to location in time and space.
The Current_Msg_Set is a set of partially ordered queues of messages. One queue for each message presentation channel. The queues contain unexpired messages delivered to the message presentation process.

An interpretive cycle, illustrated schematically in figure 3, is a convenient way to handle the management of the message sets. The parameters of messages must be used in determining the sequencing in the partially ordered Current_Msg_Set queues. An embedded reactive reasoning system use timing and priority criteria to determine the makeup of the message sets.

Message content is to be represented as suggested above in a semantic representation common to all systems. This allows diverse systems to interpret a message, it allows messages to be interpreted in different ways by different systems, and it allows message contents to be compositional. Messages can be interpreted in different ways by different subsystems. For example, in the HMI-Mgt schema it is left to the Presentation process to interpret the semantic content and format the message for the specified medium.

1.3.7 Planning Route Guidance Dialogue

The general needs for reasoning when planning the presentation of route guidance information are discussed in chapter 11, Planning Route Guidance Dialogue. We contrast the AI approach of situated planning to the more standard computer science approach of state transition networks, and discuss the advantages of each. The route guidance presentation task is first described as a situated planning problem, and then an alternative view is given using the state-transition model, which we argue is superior for this particular task. We finally show how one reasoning components necessary for constructing the route guidance dialogue is incorporated in the state-transition model.
In this paper, we focus on the reactive nature of the driver-system dialogue. The system must react to changes in the current situation that the driver is not aware of, but it must also react and adapt to the driver's actions. In itself, it is only suggestive: it can inform the driver about suitable actions, but it never performs them by itself. The reactive nature of the driver-system dialogue has several impacts on the necessary reasoning component. The most notable effect is that the dialogue must be replanned at times, the set of instructions that the driver will need cannot be foreseen in advance.

During Trip planning the HMI reasoner must fulfil two goals: Collect a complete and correct set of driver preferences. Give the driver an appropriate amount of pre-trip information about a route choice. Route Guidance, on the other hand, consists of fulfilling only one goal, that of reaching the destination. However, the system needs to maintain three requirements that affect the behaviour of the HMI reasoner: The driver must have enough information to follow the chosen route. The current position should be on the chosen route. The current route choice should be "best" with respect to the driver's preferences and the current external information (which both may change).

One particular property that we note is that the system actions often are triggered by some input: The subplans corresponding to maintenance goals are not executed because they are part of a larger plan, but because the current situation causes them to be necessary to execute. From the description of the IRG system architecture (chapter 2) we add the observation that the entire set of possible input to the HMI module is limited. This is true even for the set of possible messages from the driver, due to the limited communication channels, and the limited attention that the driver can give to the system. Finally, we note that the property that the time restrictions put on reasoning are not serious, and that the most important time requirements are instead put on the human-computer interaction.

Starting with these limitations, it is possible to describe the HMI reasoner module as a state transition system. Each state corresponds to one input to the HMI reasoner module and a related output from the module: all reasoning that needs to take place to construct that output is invisible, and for the purposes of this description, viewed as instantaneous. The states correspond to different types of waiting situations for the HMI reasoner module. The HMI reasoner module does not do any reasoning while in a waiting state (else, we might have needed tacit state transitions). In some situations, the HMI reasoner is waiting for a particular input, but the fact that time passes while the module is waiting makes it necessary to allow the HMI reasoner to accept all possible inputs in each waiting state. In [Wae 91] we will discuss the relationship between the goal-oriented description of the HMI reasoner and this state-based description more formally.

The complete set of inputs to the HMI reasoner are:

Input 1: The driver indicates that she wants to use the system. (For convenience, we assume that this only happens at the beginning of the session.)
Input 2: The route planner has generated a route.
Input 3: The driver indicates that she understands and accepts a presented route.
Input 4: The driver indicates that she does not understand a presented route.
Input 5: The driver indicates that she does not accept a presented route.
Input 6: The route monitor signals that it is time to give some particular message to the driver.
Input 7: The route monitor signals that we have gone off route.
Input 8: The meta-planner signals that the driving has started.
Input 9: The meta-planner signals that we have reached the destination.

Depending on what state the HMI reasoner is in, the action triggered will be different. In the state transition graph in Fig. 3, we can see how different events cause different state transitions. The output actions caused by these transitions are described below.

Finally, we briefly discuss one particular reasoning task we have chosen to focus upon at SICS. The system frequently needs to present a new route plan to the driver. This route plan is in most cases similar to the ones already presented. It might at a first glance seem like a
simple problem to produce such a description, but careful examination reveals some of the difficulties we have dealt with.

Figure 4: A state-transition graph describing the system functionality

The first difficulty, we note is that we must describe the new plan by relating it, not to the old route plan, but to the old route description, as that is the "common denominator" for the system’s and the users’ knowledge. Intuitively, this is due to that when the system describes the route to the driver, it does not transfer the route itself but a description of it. However, it is not convenient to store the description in the exact output syntax, for one reason because that syntax is chosen by the dialogue manager. Rather, we would like some intermediate format that is expressive enough to describe solely the information content of the messages.

The second difficulty which arises is that the new route might still fit the old description. There are two reasons for this: The initial description contained "gaps" where the driver was assumed to understand the route by himself, or the system might have postponed the presentation of some parts of the route until later (like right before they are traversed). Thus, the system must consider what to do when the new route still fits the old description. The exact action in this case depends on how other parts of the system, such as the route chunking algorithm, are constructed. For our purposes, it suffices to note that the system must be able to deduce that a new route fits an old description. For this reason, we must carefully define the semantics of the intermediate route description format.

The third difficulty is that the current route choice might have been transferred to the driver in several messages. For example, only parts of the route may have been described. Secondly, the route might have been replanned before, and consequently, the driver might already have been told about changes from the original description. Thus, the new description
must be related to the whole sequence of plan descriptions, and not only to the original, or the most recent one.

This property is actually true for all messages passing from the HMI reasoner module to the driver, and it suggests that the what module should contain a dialogue memory that mirrors the driver-system dialogue over time. Based on the state-transition model, we can model time as event histories. Each event is then characterized by the incoming message and the current state, and the resulting outgoing message(s). An event is not in the purest sense atomic, since the reasoning involved will take some time, but it is atomic in the sense that no new incoming message can abort it.
Chapter 2

A System Architecture for Interactive Route Guidance

Carl Brown, Kristina Höök, Per Lindevall, Annika Wærn
A System Architecture for Interactive Route Guidance

Working Document PS-9007

1990

Carl Brown, Kristina Höök, Per Lindevall and Annika Wærn¹

SICS Knowledge Based Systems Laboratory
Box 1263, S-164 28 KISTA
Sweden

Abstract

Interactive Route Guidance is a means of helping a driver to find her way through a city. Not only will the system guide the driver to ensure that she reaches her destination, it will also try to choose the best route that takes her there. In order to fulfil the latter obligation the system needs to consider dynamic changes in the environment, issues like weather, traffic congestions, road work, etc. These can change at any moment, even when the driver has already started the trip. Some reactivity, or replanning, is therefore needed by the system.

A conceptual system architecture for an (Dynamic) Interactive Route Guidance system (IRG) is outlined. It has been designed to fit several scenarios (on-route, off-route, etc), and is therefore open-ended enough to include new functionalities if needed. The main purpose is to coordinate the research efforts within Prometheus work at KBS towards a common system. A workbench implementation is outlined in [Lindevall & Brown].

1 Introduction

Interactive Route Guidance is a means of helping a driver to find her way through a city. Not only will the system guide the driver to ensure that she reaches her destination, it will also try to choose the best route that takes her there. In order to fulfil the latter obligation the system needs to consider dynamic changes in the environment, issues like weather, traffic congestions, road work, etc. These can change at any moment, even when the driver has already started the trip. Some reactivity, or replanning, is therefore needed by the system.

We have decided to regard an IRG system as consisting of two agents. One is the system, and the other is the driver. We have made an important restriction to the system interactions. The system is suggestive only:

- It does not affect the environment directly, and
- It only informs the driver.

With the division of the systems into two agents we can treat the driver as an intelligent, independent actor, who sometimes will understand and perform the suggested guidance instructions, and sometimes will not understand an instruction or not want to follow an instruction for various reasons. We must realize that even with the most advanced, adaptive system, that has all available data, the human driver will sometimes know more about the traffic situation at hand, and will therefore be a better judge of what to do, or will not accept to be the slave of such a system. It is therefore necessary

¹ Names appear in alphabetical order.
that the system has a more or less explicit model of the driver. We have realized the
driver model in two ways, firstly as an explicit user model, and secondly through the
way the system reacts to diversions from the route planned by the system.

In AI (Artificial Intelligence) planning literature the systems described are mostly one-
agent systems (autonomous robot systems) [Wilkins, Georgeff and Lansky, Agre and
Chapman, Rosenschein and Kaelbling, etc.], and only a few deal with multi-agent
problems [Lansky]. However, these approaches have the dynamic properties in com-
mon with the IRG task, which makes their solutions interesting to consider.

A trend in the autonomous robot area is that research have moved to a position where a
number of predefined low-level plans are used as a basis for higher level planning. Our
formulation of the route planning problem is not suited for this approach since it is only
the low level details of a route plan that are reusable. In our formulation the navigator
suggest a route-plan for the driver that effectuate this route-plan, that is, perform the
low level tasks of driving the car according to the suggestions from the navigator. If we
instead regard the two agents as one system, then the system will be following a lim-
ited number of "plans", which at the highest level are identical to the "scenarios". These
can, in turn, be realised by selecting from a number of lower-level plans. The scenarios
will dictate how the interaction between different sub-systems shall be initiated, han-
dled, and what reasoning is needed in order to make the system reactive to various sit-
uations. Our approach is therefore to first make a strategic plan (see for instance
[Wilkins]) at the route level, and then during the trip have a number of high-level sce-
narios that in some cases will require a totally new plan at the route level, and that will
guarantee us a smooth and reactive behaviour of the system.

This approach is close to what [Lansky] proposes when she claims that for multi-
agents, it will be necessary to have a strategic plan which we try to keep as correct as
possible during execution, but that sometimes will have to be totally replanned. Our
solution can be viewed as a special-purpose planner following this approach.

Lansky also proposes the use of real-time synchronizers that are inserted in the plan to
maintain correctness. They serve as a constraint satisfaction mechanism that, as
opposed to reactive planning strategies, can be given a well-defined semantics, since
we know how they affect the plan. Examples of such synchronizers in real life are
traffic lights, lines in stores and dispatch rules on the factory floor. In our case, the
planned route will serve as a continuous synchronizer.

In order to fully understand the system, we have investigated five different scenarios
that together describes the functionality we expect from an IRG. Each of them can be
'decorated' with details and variants, but in some sense they broadly cover all the vari-
sous situations we expect that the system shall be exposed to.

- The first describes the beginning of a trip, how the system is initiated.
- The second scenario describes the situation while driving, i.e. while executing a
  plan, where everything is working fine, the driver is following the plan correctly,
  and no dynamic information is coming in.
- The third scenario starts out as the first, but then some dynamic information that
  influences the current plan disturbs the trip, and we need to do replanning.
- The fourth is a variant of the second, where we possibly need to replan, but the
  reason is not dynamic information, but a deviation from the plan for some reason.
- Finally, the fifth describes the end of a trip.

The third and the fourth scenarios are similar, but the needed functionality from the
system will differ. In the third scenario replanning is always necessary, but in the
fourth, replanning might sometimes be superfluous. This is dependant upon regarding the user as an agent as indicated above.

One could argue that variants like whether we are travelling long or short distances, whether we are travelling through city or country sides, etc. should be included in these scenarios. We have chosen not to include those variants, since we believe that in the very broad description we make here, the architecture might very well cover all of those scenarios.

We start by describing the architectural structure (section 2) and then we go through each of the scenarios (section 3) indicating the data flow.

2 A proposed architecture

The architecture we propose is a system of concurrent processes, where the processes are defined from their respective competence. The set of processes are the same during all scenarios, as opposed from a more object-oriented approach, where processes could be started or terminated when needed. It is also more of a functional architecture as opposed from a task-oriented architecture.

In figure 1 we see a picture of the architecture. The arrows represent data flowing from one process to another. The dotted lines represents a triggering signal that causes the planner to replan. We start by describing each process, represented as a box in the diagram. In chapter three we discuss the scenarios, and when doing so the arrows between the processes will be given content.

![Diagram of System Architecture](image)

**Figure 1.** The system architecture.

It is advisable that input and output to the driver is integrated for all car-internal sub-system, not only the IRG sub-system, to avoid overloading the driver with information, especially in critical situations. This is further discussed in [Brown & Wäern]. The output from the IRG system does therefore not reach the driver directly (as indicated in figure 1), rather it goes via some mutual manager that coordinates the messages with messages from other sub-systems.
2.1 The meta-planning process

In the process architecture, we have included a meta-planning process. The exact functionality of the meta-planner is heavily dependent on how the different sub-systems are realized. The division of tasks and competence is not fully investigated. We have chosen to develop the individual sub-systems separately first, and then use the meta-planner to integrate them.

We have envisioned that the following functionalities are most appropriately governed by a meta-planner:

- make high-level decisions about when a new plan can be installed during a trip,
- make decision about whether a new plan is different enough from the old plan, currently installed in the system,
- what to do if the car suddenly stops (the driver might stop to eat for instance),
- handle initialization of the system,
- monitor current dynamic replanning mode (on-route, off-route etc),
- handle special execution of “parking-plan”,
- determine and handle termination.

2.2 The database

The system need several kinds of data in order to make its decisions. Some will be system internal, and some will be transmitted from the outside world. We do not make any distinction between data coming from outside world as as opposed to internal, instead, as said above, the database should be regarded as a conceptual database.

The database is divided into five different parts. The first is the static database that contains map information that is going to be used both by the planning process and the HMI process. The map information is hierarchically organised [Brown et al.]. By introducing a hierarchical structure the search-tree can be adequately pruned, speeding up the search process. We have chosen a hierarchical structure which, in some sense, reflect the map makers use of scale to enforce/reduce the level of detail. This makes it possible to use the hierarchy as a guide in constructing presentation.

The second part of the database is what we call the semi-dynamic information. It is information that is time dependent and based on statistics. Knowledge about predicted traffic density during the day is contained here.

The third part is the dynamic database that holds information about dynamic events broadcasted from some central station. We expect information about weather conditions, traffic jams, accidents, road work, etc., to be transmitted to the car.

The fourth part holds information about the vehicle; speed, acceleration, position, fuel level, etc.

Finally, the fifth part, contains user specifics, like where the trip is going, what criteria are considered important (i.e. fastest/shortest/most beautiful route etc.), does the user understand maps or not, does he/she know the area we are travelling through, or is he/she a tourist, etc. This information will be the basis of a user model that will influence various processes in the system.

2.3 The Planner

The planning module [Brown and Lindevall, Brown et al, Gustavsson and Lindevall, Lindevall91, Lindevall91b], is responsible for planning a route from a origin to a destination. The planning process is initiated by a trigger from either of the Route Monitor, the HMI or the Dynamic Monitor processes. As described above, the map is hierarchically organised and all levels of representation are isomorphic. This allows a route-
finding algorithm to be independent of the number of hierarchical levels in the map representation.

The algorithm for route-finding using the hierarchical map representation is an adaptation of heuristic search algorithms, the \( A^* \)-family of algorithms. The adaptation gives a search that intelligently disregards routes that are likely to be sub-optimal. The algorithm finds a good, but not necessarily the best or optimum route given the cost function. Different criteria for finding an optimum may be incorporated into the algorithm.

The ability to react to new information and dynamically refine a route plan are crucial factors for navigation systems in vehicles, both in terms of acceptance and usefulness. In order to enhance our route-finding algorithm with dynamic informations such as weather and traffic congestions we have developed a concept of areas. A set of areas is a parameterized, prediction of a deviation from a normal state. Each area is a parameter that indicate the amplitude of the deviation for that particular deviation type. In the route finding algorithm the heuristic is adapted to include areas when evaluating different route alternatives. The area concept is easily superimposed on the data or map structure mentioned above.

The output from the planner is a hierarchical partial plan for the trip, constrained by available resources and traffic policies, wanted by the driver. Plan refinements are interleaved with the plan execution. The planner takes into account at least the following constraints:

- user oriented, i.e. intermediate stops, time limits and so on,
- traffic policy oriented, i.e. recommended routes,
- traffic flow oriented, i.e. accidents and congestions.

We might consider to allow the user model to influence the planning as well. It is known [Elliott and Lesk, Streeter and Vitello, Höök and Karlgren] that humans chose routes that have certain properties. Properties like not containing too many turns, preferences for major roads, even if those routes will be longer than others, etc. We have investigated the possibility of allowing such properties to influence the planning, see [Lindevall and Höök].

### 2.4 The HMI

The HMI is responsible for presenting information to the user via some output channel, visual, audible or a combination. It is given the plan produced by the planner. From the plan it constructs a presentation plan and possibly explanations as to why the route was chosen, in order to convince the driver of the feasibility of the plan. The output from the HMI module is a sequence of dialogue objects, containing an action the driver needs to perform, a time when it needs to be presented (can be an interval, or a last point where the message is to be transmitted), or it might be a question to the driver that needs to be confirmed. The dialogue objects are sent to the dialogue manager [Brown and Wærn] in order to be sent to the driver.

The functionality of the HMI should be adapted to what person is using the system. This adaptation can be made more or less advanced. We regard the minimal adaptivity to be to differentiate between tourist, citizen and commuter, the reason being that these groups of drivers have completely different reasons to use the system. The tourist will use the system to find her way. The citizen will use the system to get to her destination as fast or convenient as possible, although she is able to reach the destination without the system. The commuter might be aware of all possible routes to her destination, and so she will not require a route plan from the system. Instead information about which of the known alternative routes is the best to take considering the current dynamic factors is required. This minimal user model is further discussed in [Höök and Karlgren].
A maximal adaptivity from the system to the driver, would be a personal user model that has a learning capability.

Using the driver model, we can to some extent reason about the driver goals, intentions and beliefs. This is further discussed in [Wærn91b, Wærn91c, Höök 91].

2.5 The route and dynamic monitors

The route and dynamic monitoring processes can be regarded as filters. They gather information, determines if it is relevant and then acts accordingly.

The purpose of the route monitor is to determine when we are off-route in either time or space. When that happens, it triggers the replanning process. The problem with determining when we are off-route or not, is more complicated than at first sight. Since we imagine a situation where the planned route will be presented differently to different drivers, (tourists, citizens, and commuters), the presentation will sometimes skip details we assume that the driver already knows. By details is meant parts of the route, like how to get from the drivers home to the nearest big road. The route descriptions becomes high-level and several choices can have been left open. Our problem is to decide when he is enough off-route to be lost, and when he is still following the description that was given to him. We will discuss this problem in section 3.

The route monitor is also responsible for sending information about where we are to the HMI so that it can determine how far we have got in the plan, and what to do next. We have chosen that the HMI should send orders to the route monitor to report when a certain position is reached.

The dynamic monitor on the other hand, take new dynamic information from the database and select the information that affects the current planning situation. The current planning situation is information about the drivers planning preferences, the coverage of the strategic (initial) plan in time and space. Note that dynamic information might affect some other area than the one we are travelling through, but still is relevant to alternative routes we could have taken. For this reason, the dynamic monitor need to know not only the current plan, but also something about alternative plans that could have been chosen.

The dynamic monitoring process is also responsible for sending new dynamic information to the planning process when needed for replanning.

3. System functionality via scenarios

For each scenario, we describe what every process is doing, what data it is receiving and what data it passes on to the other processes.

3.1 Initiation of a trip

Within Prometheus it has been proposed that trip-planning should be considered as a system of its own, and that it might be done at home, even before sitting in the car. We do not agree with this view for two reasons. Firstly, we believe that trip planning should be integrated with route guidance, since we shall assume that for some drivers an overview of the plan is presented already at the trip planning stage. Secondly, since we are constructing a dynamic route guidance system that will take dynamic factors into account, and the dynamic factors are time-limited, a plan produced hours before the trip might not be valid when the trip starts. We therefore assume that trip planning is done in the car, shortly before take-off. This puts some time constraints both on the planning but also on the plan presentation (if there is one before take-off).

The meta-planner triggers the HMI to initiate some of the user parameters, either by interacting with the driver (for the destination) or by delivering a set of (user-specific or pre-set) defaults. It is assumed that the starting point is equal with where the car is.
The meta-planner then triggers the route planner and a plan is produced, which is sent to all processes. The route planner will require information from the dynamic monitor and the route monitor when creating the plan. The HMI might (dependant upon user model) in this situation choose to present the plan, due to time restrictions and human perception and capability, it will have to be a very short but still concise overview. We obtain this by what we call “route chunking”. For a thorough discussion of how this can be done turn to [Höök and Karlgren].

The route monitor starts executing as soon as the driver starts the trip. It will follow the plan when doing so. The dynamic monitor can immediately start to scan the database for new dynamic information that affects the plan. The meta-planner switches from the initialization scenario to executing the on-route scenario. The HMI switches from pre-trip presentations to on-trip guidance instructions.

3.2 An on-going trip where everything is going well
The dynamic monitoring process is using the current plan to filter away irrelevant dynamic information. There is no need to know about the weather conditions in one city when we are travelling in another. The current plan used by the dynamic monitor does not need to be too detailed since it is only used to give a rough idea of what dynamic information might be important enough to trigger replanning. Two kinds of information will interesting to the dynamic monitor. Firstly, information that changes a factor that was used in the planning. Secondly, completely new information that affects the area we are travelling through. Since everything is going well in this scenario, the dynamic monitor is filtering away all dynamic information it gets as irrelevant for the current plan.

The route monitoring process is receiving information about where we are right now in terms of where on the current segment we are (possibly using dead reckoning, but basically any method that gives us good enough estimations can be used). It keeps determining that we are still on route in respect to the current plan, including time constraints. Since everything is going well, it does not need to trigger replanning.

The HMI needs to know when to send the next message to the driver, and so it requests the route monitor process (before we get on to the next segment) to send information, for example, when we are 20 seconds and 10 seconds before arrival at the next intersection. In order to do this, the route monitor needs to be able to calculate the estimated time of arrival to the next node. The HMI also has to redo the presentation plan over and over again. Some dialogue messages are time dependant, others can be displayed at any time between this segments and, say, five segments further ahead, etc. For some drivers, the HMI system will be silent, or simply display the “chunked plan” [Höök and Karlgren] on a screen.

The planning process is not doing anything, since the current plan holds and there is no need for replanning.

The meta-planning process keeps track of that the installed plan is the best one, that the driver is following it, and since this is the case, the meta-planner is inactive.

3.3 A dynamic event changes the plan
Everything is as in the second scenario, but something happens and we get dynamic information from the outside that affects our planned trip.

The dynamic monitoring process uses the current plan to decide whether the dynamic event affects us and if something must be done. It will filter away all the dynamic information that is outside the area the route goes through. It can then be made either to trigger replanning as soon as a dynamic event is affecting the plan, or it can be made more intelligent, so that it will only trigger replanning when the dynamic information is
important enough. In the first case, we can allow the meta-planning process to determine whether the new plan is different enough from the old one to be installed. In the second case, the planning process will not have to be triggered as often as in the first case. The decision on whether to make the dynamic monitor or the meta-planner more intelligent depends on how costly the planning process is.

The dynamic event can influence the plan in two ways, either it is a new event that we need to consider, or it might be a change of some of the factors that influenced the planning of the current plan. The dynamic monitor can also be made to keep track of the time duration of dynamic factors that did influence the current plan. The dynamic monitor triggers the planning process to replan, by sending all the relevant dynamic information to it.

We now have a positional problem which we have to deal with: The planner must somehow determine from where it should replan. In this particular case, it can replan some suitable distance away from the current position along the current route, or in some cases we can make the driver turn back. This “point of origin” could either be requested from the route monitor, or from the meta-planner, or calculated within the route planner just requesting the current position from the route monitor. Where the point of origin should be depends upon how fast the planning process can replan. Note also that since the dynamic monitor does not keep track of the current position of the car, the new information might be irrelevant because we already have passed the position where the information applies. This should be checked before the planner starts replanning.

The planning process makes its plan and sends it to the meta-planning process. The planning process uses also information from the static database, the semi-dynamic database and the user parameters database. It never communicates with the dynamic database itself, since we already have filtered information from the dynamic monitoring process about dynamic factors.

The meta-planning process will first notice that replanning has started. It might have the job of determining a suitable origin for the replanning session. It might also have the job of suppressing new orders for replanning.  

When a new plan has been produced, the meta-planner decides whether this new plan differs enough from the old plan to make it worth installing. If it does, it sends the new plan to all processes that use the current plan: the planner, the route and dynamic monitors, and the HMI. The meta-planner might suppress replanning until the new plan has been presented to and possibly accepted by the driver.

The HMI process is very busy during the replanning scenario. First it needs to tell the driver that dynamic factors have changed (although tourists might not be informed at all.) With the citizen and the commuter things can be done in several ways. Either the cause for replanning is presented to the driver, and the driver indicates whether he would like a new plan or not. Else, the system might explain the cause and suggest a new plan at the same time.

How this new plan is described to the driver is problematic. We might not have enough time to present the entire new plan, instead we might gain from simply switching to tourist mode. Since the driver is busy driving the car, giving a high-level citizen description of the route in this situation is quite different from the trip planning situation. If the new plan is presented, the presentation should be related to what information the

---

2 It is also possible, that the dynamic monitor should send its request for replanning to the meta-planner, instead of directly to the planner. In this case, the meta-planner could first ask the HMI to check wether the driver really wants a new plan to be formed, and then trigger replanning.
driver already has been given, for example, the HMI might describe the new plan by describing how it differs from the old plan, see [Wærn91c].

Informing the user about why replanning took place requires some knowledge about "typical events" and how they can be explained. These patterns might be different if the cause for replanning is something that has happened (an accident, for example), or something that has ceased to be true (the scene of the accident have been cleared).

Finally, the HMI process signals to the meta-planning process that the new plan can be/or cannot be installed in the rest of the system.

The route monitoring process works with the old plan until a new is installed. The "point of origin" in the new plan should coincide with some point in the old plan, and possibly, the plans will be equal for a longer distance. When a new plan arrives to the route monitor, the route monitor should trigger replanning if we already have arrived at a position contained in the old plan, that is past the part which is equal in the old and the new plan. If this is not the case, the route monitor should follow the old plan until we arrive to "the point of origin" of the new plan, and should then switch to the new plan.

3.4 Going off route

From the point of driver-system interaction, the definition of being off route depends on the mode of the driver. If the driver is a tourist, any deviation from the current plan is equivalent to being off route. With the citizen driver we get a much more complicated situation. Before the trip started the driver was given a high-level description of the route, the so-called "chunked plan" description. This description may well allow several different interpretations since we try not to be too detailed about the route. The citizen might also make his own judgements about route choices while driving. The citizen driver might also want to take another route for reasons like get a coffee break, stop to see a friend, etc.

Within the system, we must view things a little bit differently. The route monitoring process will view any deviation from the current route plan as an "off-route" situation, because as soon as the driver goes off route, the route monitor cannot act as a route monitor any longer! Still, we want the route monitor to monitor the current position (with respect to the map roads and intersections) while it is waiting for a new plan. When a new plan is provided, the route monitor will try to map the current position to the new plan. If the current position is contained in the new plan, it can start following the new plan immediately. If it is not, the route monitor might either trigger replanning, or use some heuristics that determine if we are heading for the point of origin of the new plan.

In the "off-route" situation, it is difficult to establish a suitable "point of origin" for the planner to start planning from (unless the planner is extremely quick). One way to determine an origin is that the route monitor gives the current map position to the route planner, and that the planner replans from this position. This implies that both the replanning and the presentation of the replanned route must take place within the time frame until the next intersection. Another way to determine a point of origin is to use some heuristics that estimate where we are going, and use a position along that way as point of origin.

The planning process finds out about dynamic information from the dynamic monitor, and other information from the rest of the database, and then it replans from the chosen point of origin. The new plan is sent to the meta-planner.

The HMI process firstly has to determine if the deviation was done by accident or on purpose. Conceptually, this is probably one of the most difficult situations the HMI must tackle. It has to find out what is in the drivers mind, and either adhere to the driv-
er's wishes (changing the parameters for the planning algorithm) or try to make the driver change his mind (by explaining why the installed system plan is a good one). The HMI must also establish that the driver has understood and accepted the installed (new) route plan. The ordering of these tasks is not obvious: The system might for example consider the deviation to be accidental unless the driver ignores a new plan, or it might first establish whether the deviation was accidental or on purpose (perhaps by asking the driver explicitly), and plan its further actions from this knowledge.

The meta-planning process will notice that we are off-route, and possibly indicate this to the HMI. During replanning, it might suppress new replanning triggers. When the plan has been produced, it need not compare the new plan with the old, because it is bound to be different. The new plan can be installed immediately. If the HMI indicates that the new plan is not accepted by the driver, the meta-planner might suspend all replanning until the HMI has found out the driver's wishes.

3.5 End of a trip

From a system point of view, the trip is over when it can be determined that the car is at the same hierarchical level as the destination. This mean that the partial hierarchical plan have been refined and executed down to the level of detail that correspond to or be equivalent to the level of the destination. In terms of the partial hierarchical plan this mean that the car is very close to the destination and yet it can give the driver enough freedom to park where ever he/she wants in the vicinity. The problem is to inform the driver of his/her goal relative the parking place. This point on the problem of different goals with the same trip for the two agents. The driver have a specific place/destination in mind such as the city library for example but the navigator could very well interpret this destination to mean the parking lot two blocks behind the city library. Two agents performing the same task by addressing two goals. This is basically an information burden put on the HMI.

4. Conclusions

We have outlined an architecture for a dynamic interactive route guidance system. The system includes both a trip planning facility and a route guidance facility. The main benefit of this paper is that it provides a common framework for many of the Prometheus activities in IRG at SICS/KBS. The architecture also serves as a basis for implementation of a Two Level Integration Demonstrator. For research on the HMI, turn to [Wærn91a, Wærn91b, Wærn91c, Höök and Karlgren, Höök], for research on planning turn to [Gustavsson and Lindevall, Lindevall, Brown et al.], for research on the influence of driver model on the planning [Lindevall and Höök].

The dynamic behaviour of the system is realized as a set of predefined high-level behaviours, the "scenarios". This views is closely related to the Knowledge Areas as they are used in the Procedural Reasoning System by Geogeff and Lansky [Georgeff and Lansky 87, Gergeff and Lansky 87b].

5. References


Chapter 3

Route-Finding using a Hierarchical Map Representation

Carl Brown, Annika Waern, Per Lindevall
Route-Finding using a Hierarchical Map Representation

Working Document PS-9002

Carl Brown, Annika Waern, Per Lindevall
Knowledge Based Systems Laboratory
Swedish Institute of Computer Science
Box 1263, S-16428 Kista, Sweden +46-8-752 15 00

Abstract

A hierarchical map representation is described in which all levels of representation are isomorphic. Any map object (e.g. town, region, intersection) may be described using a very limited set of primitive elements: expandable nodes, non-expandable nodes, and segments. This allows a route-finding algorithm to be independent of the number of levels in the map representation. Indeed, the number of levels in the representation is not fixed.

An algorithm for route-finding using the map representation is presented. The algorithm is an adaptation of heuristic search algorithms to a hierarchical data structure. The adaptation gives a search that intelligently disregards routes that are likely to be suboptimal. The algorithm finds a good route (optimal) but not necessarily the best (optimum) route. Different criteria for optimization may be incorporated in the algorithm.

The combination of map representation and route-finding algorithm allows the user to use specific or general descriptions of either or both of the starting point and destination point. A starting point might be, for example, a city name while a destination might be a particular point on a road.

1 Introduction

In simple terms, the problem for the driver is to find a way from one place to another. The driver using an interactive route guidance (IRG) system expects a description of the route.

This paper addresses one aspect of the IRG problem in Prometheus, namely the representation of information about routes, and the use of that representation for route-finding. The traditional road map contains a great deal of information. We have emulated that information in our conceptual view of knowledge representation for routes using the hierarchical nature of information found in route maps. In addition, we have designed the knowledge representation so that dynamic information about route segments and nodes can be included. In this way the navigation system will be able to select routes on the basis of these dynamic factors as well as on the basis of distance to be covered or time necessary. In modern traffic management, dynamic factors such as pollution and traffic congestion must be considered in route-finding algorithms.

The internal representation of route knowledge in the navigation system we describe can be used to aid in the presentation of route information to the driver at different levels of detail.
We do not deal with such presentations in this\(^1\) paper but the use and needs of such presentations has been a consideration in the work.

Another consideration in the design of our route knowledge representation has been the use of hierarchical information in the dynamic replanning of routes. In our dynamic replanning system, the subject of current research, replanning can be guided, in part, by the hierarchical nature of the knowledge representation.

In section 2 we examine the knowledge we wish to encode for IRG and Dynamic RePlanning of routes while section 3 goes into detail about the knowledge representation. The use of the representation is discussed in section 4 where we present an algorithm for route finding. In section 5 we present empirical results which demonstrate that our representation and algorithm give acceptable search times. Sections 6 and 7 discuss our conclusions and ongoing work.

2 Knowledge needs for route-finding

Traditional route maps contain a great deal of information. The most obvious feature is that they are structured. The structure is evidenced in the nature of the illustration of a road. Good roads are shown as wider lines, or perhaps with special colour combinations. The information conveyed to the user of the map is that such a road will have a high traffic capacity, perhaps a high speed limit, and generally fewer intersections. Routes with narrow lines convey the opposite. The mapmaker has structured the information in the map. Road types are explicit. Road capacities are implicit.

Another function of many route maps is to give an overview of an area. The user can perceive regions of a map. This is accomplished consciously by the designer of a map through the choice of level of detail. In fact, a map represents a single hierarchical level of information. For example, a map at a scale of 1:200 000 depicts routes between urban centers, but no details of routes within those centers. A map at such a scale depicts routes and regions appropriate to that scale and appropriate to a certain level of route finding. To find routes within the regions or urban centers of such a map, one must "descend" in the hierarchy of maps to a level of higher detail.

One feature of maps cannot be captured in the knowledge representation. Maps provide hints about which roads are shortest, fastest, or "best" to get to a destination. This is, of course, an interpretation of the map. We deal with this in the section 3 on route-finding, where an algorithm is developed to interpret the map representation and find an optimal route. Optimal in this sense means the best route with respect to a set of objectives.

The objectives will govern the interpretation of the map. Such objectives may be any one or a combination of the following: shortest route, minimum fuel consumption, minimum contribution to pollution, shortest expected elapsed time, avoiding polluted areas, avoiding congested areas, most scenic route, avoiding road repairs in progress, avoiding traffic accidents, etcetera. Clearly some of these objectives are affected by dynamic factors. Route segments may be congested at some times of day and not at others, or on working days and not on weekends. A region may be heavily polluted one day and not another. Weather conditions may affect the capacity of a road segment, as may traffic accidents and road repairs. Short or long term dynamic factors such as these must be considered in meeting the objectives.

\(^1\) Forthcoming publications will present criteria and methods of presenting route information to the driver using the hierarchical route information described in this paper. [A. Waern, SICS-KBS PS-9001].
The SMRF\textsuperscript{1} knowledge representation we have developed captures the hierarchical nature of traditional road maps. It makes it possible to take an overview of regions and urban areas. Details not important to the search for an optimal route are hidden in the hierarchy until needed in the search. It is possible\textsuperscript{2} to include dynamic and static factors in the interpretation of the SMRF. Dynamic factors will be tested in our ongoing research programme.

Additional information to enhance route presentation (restaurants, petrol stations, shopping centers, etc) may also be encoded in a knowledge representation for maps.

We now turn to a description of SMRF and how it is used by a search algorithm.

3 The hierarchical map representation SMRF

The SMRF hierarchical representation of a map is conceptually a directed graph with costs assigned to the arcs of the graph. The cost assigned to an arc represents the "cost of traversal" of the arc. Such a cost represents total cost of static and dynamic factors for that arc, summed appropriately according to the current objectives. We discuss cost of traversing arcs and the incorporation of these factors in sections 5 and 7.

The hierarchy is based on the assumption that main roads with a presumable higher speed limit and higher degree of maintenance are both shorter, safer and faster than regional roads under normal circumstances. This assumption holds unless the regional roads could be a short cut or the main roads are obstructed for some reason. To mirror this assumption, suitable regions of regional roads are clustered into a node, presenting a whole region of roads as a single node. This node is then given a suitable weight/cost estimate to be used by the search algorithm to decide whether or not it should be included in the path. At a regional level new regional nodes can be encountered and the procedure is repeated. The same hierarchical treatment applies to cities. Once a regional node is included in the path it must be "opened up" to see if the weight/cost estimate is correct and to find the correct path to traverse it.

\textsuperscript{1} SMRF = SICS Map Representation Format
\textsuperscript{2} Currently we have included only the static factors, road segment length and geographic location of nodes, in our testing of the route-finding algorithm. See section 6.
Figure 1. A Road Map of The Country

This kind of complex node will subsequently be called an expandable node. Figures 1, 2 and 3 illustrate the use of expandable nodes.

Figure 2. The Country as Nodes and Segments.

Figure 3. Big City Expanded

Figure 1 is an ordinary road map of an imaginary country. Figure 2 is the same "country" but represented as nodes (both expandable and non-expandable) and segments. This is also the highest hierarchical level of the database. Figure 3 is an expansion of the expandable node big_city in figure 2. Figure 3 is an example of expandable nodes containing expandable nodes.

For algorithmic reasons, we require that segments from an expandable node must lead to a non-expandable node. This means that two expandable nodes can not be connected by a segment directly but must have at least one non-expandable node in between.

The smallest entity of the hierarchy is the intersection node. The intersection node is non-expandable and can appear at any hierarchic level. A final search path is found when all the nodes in the path are intersection nodes. All weight/cost estimates in the regional/city nodes have then been replaced by actual costs. Intersections might be further expanded in a presentation node (presented as a node & segment path or as a temporal dialog corresponding to entry-exit points of the intersection).

Each node have a route-list, a list of possible "routes" through the node. The purpose of the route-list is to establish legal connections through the node. A "route" consists of an entry-segment-name, an exit-segment-name and a cost-estimate for the "route" in question. By letting nodes contain information on how it could be traversed we have accomplished two things. First, illegal or non-existent routes through an intersection can be handled by the search algorithm at the same time it is looking for successors. Knowing the entry segment the successor must be an end node of a corresponding exit segment. Second, in "expandable" nodes the cost estimate covers an expected path through the hidden region while the cost estimate in intersections are simply the cost for turning left or right or to go straight ahead. Cost estimates of intersections is a god way to model that left hand turns can be more costly than right hand turns. The use of exandable nodes will be further explained in section 3.3.
Figure 4. Possible routes through an intersection.

Figure 4 is a map or a visual instruction of possible routes through an intersection. The route list in intersections is an abstraction of figure 4.

The node name is the key to the hierarchic structure. Search is performed from a start node to a goal node. Assume the following initial start and goal node.

\[
\text{StartNode} : \text{country, small\_town, city\_center} \\
\text{GoalNode} : \text{country, big\_city, down\_town, main\_capote}
\]

The convention used in node names is that the last entity of the name set is the node name and its predecessors in the name set are an address. This means that the name of the destination node is main\_capote and the address is the set \{country, big\_city, down\_town\}. This address is itself a node name at a higher hierarchic level whose address is the set \{country, big\_city\} and which in turn is a node at the highest level of the hierarchy. The top level address is \{country\} and a top level node name is \{country, node\_name\} (a set of 2 objects).

Segments have, in addition to an address and a name (road/street name), a sequence number. A segment name at the top level of the map hierarchy is thus a set of 3 objects; \{country, name, number\}. Note that the number of objects in the name set indicate the present level in the hierarchy.

4 Routefinding in a hierarchy - an algorithm

If our hierarchical map representation is an interpretation of how the mapmaker structure map information, we would like to see the search algorithm mimic a person planning trips with the help of a map. In order to do so, we need some heuristic to guide the search towards the destination rather than an exhaustive search of the hole map.

4.1 Heuristic search

The search algorithm uses heuristics to find a short path between a start node and a goal node. The algorithm is basically the A* algorithm [Hart68],[Rich83], but the hierarchic map representation and the estimates of the expandable nodes cannot assure an optimal path. Thus we have sacrificed optimality for efficiency.

The A* algorithm use a cost estimate to decide what path is most likely to lead to the goal node. This cost estimate, f', consists of two parts, a true cost g and a heuristic h'. The true cost, g, is the sum of segment costs from the start node to the current node. The heristic, h', an estimated remaining cost of getting from the current node to the goal node. The heuristics used is the straight line distance from the current node to the goal node.
Our search algorithm keep a set of candidate paths, together with their true cost $g$, in a set. The same cost estimate as the $A^*$ is used. To avoid that the search algorithm is getting tangled into loops and to minimize the number of candidate paths the algorithm uses two sets to keep track of "passed" and "visited" nodes. The set of "passed" nodes contain nodes that have been examined for successor nodes. The set of "visited" nodes contain nodes that have not yet been examined for successors but may be. The first thing the algorithm does when traversing a node is to check if the successor node is contained in the list of "passed" nodes. If it is a member the successor node will be discarded. If it is not a member, we have found a new path candidate. The traversed node is moved from the set of "visited" nodes to the set of "passed" nodes. Secondly the search algorithm looks through the list of "visited" nodes for duplicates. If duplicates are found the path costs of the duplicates are compared and the duplicate with the lowest cost are left in the "visited" set.

The search algorithm can be described in the following steps.

Step 1. IF the current node of the chosen path is the goal THEN terminate the search.

Step 2. ELSE Find successors to the current node in the chosen path and add the current node to the set of passed nodes.

Step 3. Discard those successors that are members of the set of passed nodes.

Step 4. Construct new paths, with the successors as their current nodes.

Step 5. Add the new paths to the set of possible paths.

Step 6. Sort the set of possible paths on best cost estimate.

Step 7. Pick the path with the lowest cost estimate as the new chosen path and repeat from step 1.

Expandable nodes "hide" entire road maps that have many possible entry and exit points. There can be many different paths that connect the entry-exit pair. For this reason an expandable node can not be discarded after the first time it has been passed or visited. Thus only non-expandable nodes can be added to the set of passed nodes, and in the set of visited nodes, expandable nodes can appear several times.

4.2 Describing the Start and Destination

Initially the search algorithm look for successors to the start node only on the same hierarchical level or on the next level above. To terminate the search the goal node have to be on the same level as the start node or on a higher level. This is not always the case. But the naming convention, with names of expandable nodes being the address to all nodes hidden beneath it assures that the names of all expandable nodes above the goal node is actually contained in name of the goal node, and can be used for intermediate goals by the search algorithm. Thus if the goal found is only an intermediate goal the search have to continue on a lower level, and have to do so until the original goal node is encountered.

After the first search the path may include several expandable nodes. To expand a path one has to find a path on a lower hierarchical level given the entry and exit points to this level from the level above, that is, a search at a lower level where the start node is the entry point and the goal node is the exit point to this region. The path found at the lower level is inserted in the previous unexpanded path and the cost recalculated.

4.3 Expandable nodes in the heuristic search

In this section we are going to describe in more detail the primitives that accomplish the hierarchic structure in the map database and some aspects of the search algorithm.
To explain the role of segments and the nodes of type "expandable" we will use an example of a successful search from m2_1 to m4_1 in "the country". The example is illustrated in figure 5. The full name of the start node is \{country, m2_1\} and for the goal node \{country, m4_1\}. Both start and end nodes are nodes of type "intersection" at level 1.

Figure 5. Showing a route over an expandable node and it’s hidden structure.

The algorithm start out by looking for successors to m2_1 at level 1 and find (among others) a node of "expandable" type called \{country, central_region\}. It adds this node to the path that becomes:

\{country, central_region\}, \{country, m2_1\} and have the cost 0.

The cost is at this stage 0 (zero) because the segment's LENGTH is 0. If we assume that this path is chosen as the best to proceed the search from, the algorithm will go on to find successors to \{country, central_region\}. To accomplish this it examines the route list of \{country, central_region\} to find out which nodes are reachable from central_region coming from m2_1. In this particular case, all the nodes bordering to the central_region are reachable from m2_1 through central_region. One of these is our goal node \{country, m4_1\}. The goal node is added to the path that now look like:

\{country, m4_1\}, \{country, central_region\}, \{country, m2_1\}

and the cost estimate from the route list 106.45 is added to the previous cost.

Now that the search has reached the goal the algorithm checks the path for expandable nodes. As central_region is an expandable node, the algorithm has to find a path at a lower level that connects the two border nodes to central_region. These becomes start and goal nodes for the next search pass. The algorithm also know that the name of the expandable node is the address to all nodes of the region hidden under this expandable node. To find out where to start and on what level, the algorithm goes back to the predecessor of central_region (in this case m2_1) and looks for a successor to m2_1 with the address \{country, central_region\}. This node is \{country, central_region, west_junction\}.

Now, the algorithm can proceed at lower level, starting with the path:

\{country, central_region, west_junction\}, \{country, m2_1\} whose cost is 24.0

The search continue as before and look for successors to the node \{country, central_region, west_junction\}. The best successor is \{country, central_region, central_junction\} and that this successor connects to m4_1. As this path does not contain any expandable nodes the final path will be:
{country, m4_1}, {country, central_region, central_junction}, {country, central_region, west_junction}, {country, m2_1} and the cost is 106.45

The route list in a node thus plays a crucial part in keeping the algorithm on the right hierarchic level. If an algorithm did not consult the route list the same map database would be entirely flat and have a set of nodes (the expandable ones) and segments that were cheap short cuts over large regions.

Coordinates for non-expandable nodes should be their geographical location. For expandable nodes the most obvious coordinates should be used, or the geometric mean of the coordinates of all nodes in the region.

5 Results

Tests have been carried out to establish performance differences between SMRF and the A* algorithm using a flat version of our map database. 128 different runs were constructed representing six different situations of operation of the SMRF. The six different situations are:

1. Paths that has been found on one hierarchichal level.
2. A path going down the hierarchy and up again.
3. A path from a lower hierarchichal level to a higher.
4. A path from a higher hierarchichal level to a lower.
5. A path going up through the hierarchy and down again.
6. Pathes that goes up and down or vise versa several times before the goal is reached. In general "long tripps" traversing several regions.

Diagram 1. Search Space vs Search Time

5.2 The Different Situations Explained.

The first situation: A* is allowed to excel here simply because A* (with less computational overhead) is faster then SMRF. The reason for this is that most of the search is done in areas of the graph with uniform segment cost (length). In the runs were SMRF is faster the search has began on the "border" to a region that contain segments with a magnitude smaller costs. When A* start its search it find an segment with a small cost, and this path turns out to have
the best total cost. A* is now trapped in the fine segment region and the only chance to come out of this region is to collect a path (in the wrong direction) that have a larger total cost than the path that leads to the goal. As the segments in fine region have a small cost assoicated to them A* have to travers many nodes (build a large search space) in order to build up a total cost that can direct the search to the correct path. SMRF does not encounter this problem because the fine segment region is hidden under an "expandable" node and if this node ever will be included in the final path it is not "expanded" until the final goal has been reached. Even in this situation SMRF would perform better than A* when investigating a fine region. This is because the search (when "expanding") is done between two border nodes to the fine segment region which lets the heuristic cost to work with the same order of magnitude as segment cost. This leads us into the discussion of the second situation.

![Diagram 2. A* Search Space vs Search Time](image)

The second situation: Here is a situation that simulate traversing an "expandable" node, that is, traversing an area of usually smaller segment cost than the algorithm have been exposed to so far. If the map implementation has been done properly and dynamic effects such as road work or traffic congestions do not throw the estimated cost off, SMRF will look for a path only one hierarchical level down. Thus no bigger performance gains can be expected from hidden neighbouring regions at each succesive level. A* perform best in the situations where the fine segment area is close to a straight line connecting start and goal, that is the fine segment area must be at the begining or at the end of the best path or the best path must be close to this straight line. This has to do with the heuristics used. Being on the connecting straight line any deviation from it is recorded by the heuristic function and added to the cost estimate, but to move along the straight line will have no or little effect on the cost estimate. If the fine segment area (on the best path) is located a bit on the side of the straight line connecting start and goal, then the heuristics is more prone to investigate side tracks.

The third situation: As the SMRF is setup to automatic increase hierarchical hight there is almost no differens algorithmically between A* and SMRF in this search situation. Performance is in general within 15% of each other. In one instance A* got entangled in a fine segment area when investigating a path that was not choosen as the final path and a situation similar to the one described above developed. Here SMRF proved itself by being 4.4 times faster than A*. The situation is that the search algorithm has two different prime paths to choose from as its best candidate path and one of them has a region of fine segments.
The fourth situation: When going down the hierarchy SMRF search one hierarchical level at the time. This provides the possibilities of performance gains by discarding expandable regions at the same hierarchical level. A situation often encountered within cities. In situations were SMRF perform better the goal is located in lower hierarchical regions that are large and have several neighboring regions of fair size at the same hierarchical level. Performance for SMRF is around 20-30% lower than A*. The runs were A* perform better have the goal node at the lower level very close (both in terms of cost and number of traversed nodes) to the entry node of the top hierarchical level. The two prime path situation described above (reversed order of traversal) applies here as well but SMRF being almost 9 times faster.
The fifth situation: This situation is a combination of the third and fourth situation. It will apply to tripps between cities or two suburbs in a city. Here SMRF is quiet consistantly performing 20-50% better than A*. The only time A* is better is when the found best path is close to the straight line between start and goal. In cities traffic policies, such as one way streets, restricted admittance areas, force the traffic into special detour links, a situation that does not favour A*.

![Diagram 5. General Performance vs Path Length](image)

The sixth situation: As a combination of all previous situations it is not surprising that SMRF really come to its best here. Passing more than one fine segment area on its way, SMRF never consider to investigate these areas any closer until the goal is reached, while A* has all the disadvantages described above.

![Diagram 6. General Performance vs Situations](image)
6 Conclusions

The SMRF have accomplished what we set out to do, to assist the search algorithm in getting an broad overview of the situation before it go down into details. By providing an overview SMRF enhance efficiency of the search algorithm. Two notable conclusions can be drawn from diagrams 1 through 6. First: SMRF perform more uniform than A*. Diagrams 3 and 4 show that variation between SMRF runs is smaller than the corresponding A* runs. In general SMRF use less search space and thus less search time than A*. On average the search space factor (A* search space/SMRF search space) is 2.2 for the tests to date but as high ratios as 10.5 have been recorded. Second: The longer the path the better SMRF can be expected to perform. Diagrams 5 and 6 clearly show that an increase in path length (increase in search space) is favourable for SMRF. The only situations where SMRF could not be expected to do better than A* is short paths and situations were the search space do not increase for A* (the 1th situation above).

7 Ongoing work

Static cost is what normally applies to graph theory. Dynamic cost on the other hand have many different forms and representations. Dynamic costs can be regarded as static in a planning situation simply by freezing them at the instant the planning begin. To get the most out of the dynamics we propose to look at instant cost at the time the planning wavefront pass the node or segment in question. Thus temporal representation is crucial to dynamic costs.

Dynamic costs have a dual nature as they are individual to each node and segment and at the same time applies to larger areas. What we have in mind is that for instance a segment have an expected traffic capacity that varies over the day and with the day of the week, but the weather condition that applies to a large area, effects all nodes and segments in this area. Even sudden changes to the traffic situation, for instance accidents have an impact on surrounding nodes and segments, that is, applies to an area rather than individual nodes or segments.

For the ongoing work we are looking into how different aspects of static and dynamic costs are weighted into the cost estimate and for suitable dynamic heuristics.

Other types of search strategies require further investigations on search algorithms. The here mentioned hierarchical representation is suitable for bidirectional search schemes and thus to parallel processing.

We are also developing tools for mapping ordinary road maps, that is structured in political or economical hierarchies, into our representation.

References


Chapter 4

Planning with Dynamic Parameters

Rune Gustavsson and Per Lindevall
Planing with Dynamic Parameters

Rune Gustavsson and Per Lindevall

Knowledge Based Systems Laboratory
Swedish Institute of Computer Science
Box 1263, S-164 28 KISTA, Sweden  +46-8-752 15 00

Introduction

We have developed an algorithm for route-finding using a hierarchical map representation [Bro90]. The algorithm uses only static information, i.e. path length, as a parameter in its cost function. It is however possible to add dynamic parameters, into the cost function and thus enhance the competence of the resulting navigation system.

In another paper [Gus89] we have addressed some basic issues in the area of dynamic planning and traffic flow, advocating different parameterized models of the traffic flow.

In this paper we describe a general knowledge structure for incorporating new dynamic information with old dynamic information. A notation for traffic flow parameters was introduced in [Gus89] in which an index k on the traffic-flow parameters is used to denote influences of weather etc. By marking each segment of the route with a k-value, an expected travel time for the segment in question can be calculated from the k-dependent traffic-flow parameters. Traffic flow is assumed to be statistical/historical data for the segment under various conditions. This implies that traffic conditions have been classified somehow. The idea is to have a default k-value that applies to normal traffic conditions and abnormal conditions, with a different k-value, will overrule the normal situation.

Figure 1. The Country as a graph of segments, intersections nodes and expandable nodes.
Route-Finding using a Hierarchical Map Representation

In [Bro90] we introduced a hierarchical map representation in which all levels of representation are isomorphic. Any map object (e.g., town, region, intersection) may be described using a very limited set of primitive elements: expandable nodes, non-expandable nodes and segments. This allows a route-finding algorithm to be independent of the number of levels in the map representation. Indeed, the number of levels in the representation is not fixed.

The algorithm for route-finding using our map representation as an adaptation of heuristic search algorithms, the *-family of algorithms, to our hierarchical data structure. The adaptation gives a search that intelligently disregards routes that are likely to be sub-optimal. The algorithm finds a good, but not necessarily the best or optimum route given the cost function. Different criteria for finding an optimum may be incorporated into the algorithm.

Our combination of map representation and route-finding algorithm allows the user, driver, to use specific or general descriptions of either or both of the starting point and destination point. A starting point might be, for example, a city name while the destination might be a particular point on a road.

The heuristic search algorithm builds a search tree while finding a good route from start to destination. We have for instance Figure 2 below, with notations from Figure 1.

![Search Tree Diagram](image)

**Figure 2.** A search tree for a trip between `boom_town` and `big_city`.

Dynamic Planning

In our paper [Gus89] we addressed some basic issues in the area of dynamic planning. The ability of and use of dynamic replanning is a crucial factor for the acceptance and usefulness of navigation systems in vehicles. We take, at this point, granted the existence of an Interactive Planning System IPS embedded as shown in figure 3.

We assume that the output from the planner is a hierarchical partial plan for the trip, constrained by available resources and traffic policies, wanted by the driver. We furthermore assume that the plan refinements are interleaved with the plan execution and that relevant information is communicated between driver and planner. The monitor also checks that relevant preconditions for the execution of the task at hand are fulfilled. The IPS planner takes into account at least the following constraints:

- user oriented, i.e. intermediate stops, time limits and so on
- traffic policy oriented, i.e. recommended routes
- traffic flow oriented, i.e. accidents and congestions
We concentrate the following discussions on the types and impacts of the traffic flow oriented constraints for the IPS.

![Diagram of IPS system](image)

Figure 3. Outline of an embedded Interactive Planning System, IPS

The output of the planner is a hierarchical partial plan as depicted in figure 4. For instance at the route level the plan is an augmented directed graph, where the arcs consists of the parts of the route where one particular action such as *follow* is valid, i.e. *follow E4*. It is important to note that this level of the plan corresponds to a level structure of the information and dialogue model for the trip. A replanning and rerouting due to an expected congestion 20 km or 20 minutes away should be advised to the driver with appropriate explanations. The channel and level of explanation for an advice on the lane level is quite different although the message, *turn left*, could be the same.

![Hierarchical partial plan](image)

Figure 4. A hierarchical partial plan
Dynamic Parameters

The circles at the route level in figure 4, correspond to major road intersections, towns or cities. They correspond to expandable nodes of our route-finding algorithm mentioned above.

The function of the Monitor in Figure 3 are to perform the following four tasks

- Monitoring the plan at all levels
- Initiate plan refinements
- Detect deviations from the original plan
- Initiate proper actions

We briefly define the sub tasks involved performed of the Monitor in the following way. It is essential, both for the replanning and for the communication, that the monitoring is performed on every level of the hierarchical plan. Some of the monitoring tasks are

- checking that the vehicle "follows the right segment", or more generally
- checking that the preconditions for the actions performed are still valid.

Some of the preconditions are estimates of time, traffic flow, remaining part of the plan P. Due to the dynamic situations at hand the plan refinements should be performed as late as possible. The i\(^{th}\) level of the plan is responsible for initiating plan refinement of level i+1. The detection of deviations from the plan under execution is essential. Essentially the task is to derive, from sensor inputs, that a deviation has or will happen and then to perform proper actions. We have, in an other paper, used Petri Nets to model classification of situations and to model plans for recovery. This idea should be elaborated and tested further.

Traffic Flow and Dynamic Parameters

A segment i\(^{\|}\) of the graph has the following traffic flow parameters

\[ l_{ij} = (\Delta t^i_{ij}(t), d^{ij}, F_k^{ij}(t), R_k^{ij}(t), C_k^{ij}), \quad \text{where} \]

- \( i, j \) denotes the corresponding arc in the directed graph, where i and j designate nodes
- \( k \) is an index corresponding to the influence of dynamic factors on the traffic flow parameter
- \( t \) denotes the time, i.e \( t = (\text{month, day, hour, minute}) \), or a relevant subvector of it
- \( d^{ij} \) is the length, in km, of i\(^{\|}\)
- \( \Delta t^i_{ij}(t) \) denotes the expected travel time at time \( t \) of the road segment i\(^{\|}\)
- \( F_k^{ij}(t) \) is the traffic flow at time \( t \) expressed as km/h
- \( R_k^{ij}(t) \) is the density of vehicles at time \( t \) expressed as vehicles/km
- \( C_k^{ij} \) is the capacity of the segment i\(^{\|}\) expressed as vehicles/h

The dynamic factors used to determine the particular k-value include weather, congestions, atmospheric pollution and traffic accidents. We note that political decisions about desired traffic flow may also influence the choice of k for a route segment i\(^{\|}\).

Dynamic characteristics of a segment is based on static traffic flow information. With knowledge of the conditions under which the information was gathered k-values can be assigned to a set of traffic flow curves. Each k-valued flow curve will then describe the traffic flow as a function of time-of-day-and-year for an expected condition. With knowledge of the expected traffic flow the travel time for the segment can be found. Traffic flow is traffic density R as a function of time t and traffic mean speed F as a function of traffic density R. See figure 5.

For traffic control purposes areas can be assigned unfavourable k-values to suppress the use of these areas. The traffic control centre must then know what k-value to use to get the
expected behaviour from the traffic. Only cars that use any form of dynamic parameters for costs in there route optimization's strategy will be effected by the traffic controllers messages. Area avoidance can be built in to the heuristics such that messages from the traffic controllers will always be considered even if the planned route is an optimum on static parameters.

The traffic density curves in figure 5, have the nice feature to be able to predict how time-critical a segment is and thus in the end a whole path. The time derivation of the traffic density at time t determine how vulnerable the segment is for being on time. This is a good measure for reasoning about timing constraints for paths.

![Figure 5. Density R as a function of time t and the Mean Speed F as a function of density R.](image)

We furthermore assume that the following relation (1) holds in a suitable open domain $W_{ij}$

$$ F_{k}^{ij}(t) R_{k}^{ij}(t) = C_{k}^{ij} \text{, when } (k, t, F, R, C) \in W_{ij} $$  \hspace{1cm} (1)

From a given set of parameters we can calculate some of the others, i.e.

$$ F_{k}^{ij}(t) = \text{minimum}( F_{0}^{ij}(t), C_{k}^{ij}(R) / R_{k}^{ij}(t)) \text{ km/h} $$

$$ \Delta t^{ij}(t) = d^{ij} / F_{k}^{ij}(t) \text{ h} $$  \hspace{1cm} (2)

where $F_{0}^{ij}(t)$ is the maximum allowed speed for the road segment $i^{j}$, that is, the speed limit. and $\Delta t^{ij}(t)$ is the expected travel time of $i^{j}$ if the trip starts at time t.

If the planned trip P consists of the road segments $i^{i}, i+1, i = 0,...,n-1$, then we have

$$ d_{p} = \sum_{i} d^{i,i+1}, \text{ the total distance of the trip } P $$  \hspace{1cm} (3)

$$ t_{p} (t) = \sum_{i} \Delta t^{i,i+1}(t_{i}), \text{ the expected time to travel the trip } P $$  \hspace{1cm} (4)

$$ F_{p}(t) = d_{p} / t_{p} (t), \text{ the expected mean speed when traveling P } $$  \hspace{1cm} (5)

The equations (1) to (5) are inputs to the IPS planner as parameters for generating an optimum plan, by some given measure or cost function.

The choice of the index k for a segment's parameters determines the cost of traversal of that segment. Thus the choice of the k-values will influence the route planning and to make sense the choice of k-value for segments must be related to the dynamic factors mentioned above.
A Representation of Dynamic parameters with Areas

In order to enhance our route-finding algorithm with dynamic parameters such as weather and traffic congestions we have to superimpose these informations on the data or map structure defined in [Bro90] mentioned above and to adjust the route-finding algorithm accordingly.

A natural way to superimpose dynamic parameters such as weather is to define (dynamic) areas corresponding to the various, parameterized, weather situations. If a chosen route is intersecting such an area the corresponding cost function is updated and an eventual replanning has to be considered. The concept of dynamic parameters as areas makes the adjustment of our route-finding algorithm feasible as we will see later.

With each area, that is a closed polygon, we associate the proper k-value in accordance with our discussion under Traffic Flow and Dynamic Parameters above. By incorporating the travel time in the cost of traversing segments, the effects of the weather will be considered during planning for a route. By optimizing travel time during planning the worst affected segments will be avoided by our route-finding algorithm.

In the same vein we could use areas for representing such things as polluted areas, traffic congestions etc. By associating the areas with unfavourable k-values (parameters), given for instance by the traffic controllers, we can make our route-finding algorithm avoid these areas at planning time. Pollution is formed and spread in a similar manner as weather, in fact it is weather dependent. That is the area representation is natural for phenomena such as pollution.

Traffic congestions on the other hand is due to sudden expected or unexpected changes on a single segment or a joint between two segments. This could be things like road work, two lane traffic yield into one lane, speed limit changes, high traffic density in general, accidents etc. The point here is that any of the mentioned causes are traffic density dependent and as such, when "critical mass" have been reached, will effect not only the segment with the actual road work etc but rather all joining segments to varying degrees. This mean that a whole group of connected segments, spread over an area, will be congested due to the road work and the concept of area representation is applicable.

It seems even plausible that the concept of areas can even be used in order to optimize some kinds of scenic routes. One way to do it, and still let the driver choose what a scenic view is, is to divide the country into small areas labelled with their typical cultural features such as flat farm land, 1700-th century city centre, 1960-th suburb, hilly pine forests, mountain range etc. In this way routes can be maximized to the drivers general liking but specific tourist attractions such as Leckö Castle or a museum tour could not be represented or planned with areas.

Area Representation and Route-Finding

When searching for a path or actually planning for a trip one is concerned with two different measure of the current path to be able to judge it's expected performance against other paths that one know about. These two measures can be described as "what has happened in the past" and "what can be expected in the future". The former measure is known and simply a summation of the encountered costs of the path up to the current position. The measure is called the cost function. The latter measurement deals with the future or the reminder of the path and when based on some kind of common sense the prediction is called heuristics or heuristic function. In a means ends analysis of the total cost or prediction of a path, the total is simply the sum of the value from the cost function and the value from the heuristic function. Classes of such search algorithms are investigated in different areas of AI. A well-known class is the A* algorithms, see [Bro90].

We remark that we in this context only refer to algorithms which can be used in automatic planning systems. Actual methods or algorithms used by human drivers are sometimes of quite different nature. An obvious example are methods for short range route finding found in papers by Streeter & Vitello [Str86] and by Elliott & Lesk [Ell82].

Dynamic events represented as areas and their associated k-values are used to determine the cost to traverse a known segment at some time t (see the discussion under Traffic Flow and Dynamic Parameters above). Each area entering the IRG system have to be mapped to the
corresponding nodes and segments in the database. In order to bring down the amount of work involved to label each node and segment with a proper k-value the hierarchical representation used in SMRF [Bro90] can be used to infer k-values for hierarchical sub-trees. The idea is that if an area can be shown to cover a whole region (an expandable node and the set of intersection nodes that connects to that expandable node) than all intersection nodes and expandable nodes that is covered by the region (hierarchically below that expandable node) must also be covered by the same area.

A small example using figure 6 as illustration to show the principles. It is assumed here that we can determine if the segment end nodes are inside or outside the area in question. If only one of the end nodes are inside the area the segment will receive the k-value of the node that is outside the area. In the situation in figure 6 below (for comparison see figure 1) we assign k-value 9 to the segments between main_junction and m4_2 and between main_junction and m3_1 despite that m4_2 and m3_1 have k-value 8. The lower level segment between m3_1 and m4_2 within eastern_region will receive a k-value of 8.

![Figure 6. The Country with an area.](image)

**Route-finding Algorithms Adopting Areas for Dynamic Parameters**

Our adaptation, [Bro90], of the A* algorithms to a Route-finding algorithm uses the following cost function when searching for best continuations of the planned route so far

\[
\text{Total Sum} = \text{Cost} + \text{Heuristic}, \quad \text{where}
\]

Cost is the actual cost of the route planned up to this point and Heuristic is an estimate of the costs of the possible continuations.

To cope with dynamic parameters we propose an optimization strategy that uses a heuristic head. That is the new cost function can be expressed as
Total Sum = Cost + Heuristic (1 + Heuristic Head)

The Heuristic Head (HH) is a factor associated with areas that are assumed that the path will traverse on its way to the destination. Areas of easy access have a low HH while areas of high impedance have a high HH. A path that consider crossing an unfavourable area will receive a higher total cost than the same path would have received if the area was not considered. The heuristic function Total Sum can be explained as a terrain navigation or obstacle avoidance procedure.

From the current position on a selected path between start and destination the heuristic investigates if there is any area of interest between here and the destination. As the planner don't know exactly how the potential future path will look from the current position to the destination it must simply assume something about what can be encountered. One common way to do this is to assume straight line distance (the double headed arrow in figure 7), perhaps with some weighting factor representative for the "quality" of the chosen path, or if in a city, a manhattan distance of some orientation. If the planned path crosses an area a weight factor or HH is calculated based on the amount of overlap, area content and k-value.

Figure 7. An illustration of dynamic planning.

As the computational load for the heuristic head, HH, is expected to be quite high or repeated many times we suggest the use of a heuristic set which is a set of heuristic parameters derived each time the HH is calculated. A suitable collection of the members in this set is then used to determine a HH for various optimization's criteria such as travel economy, global and/or local impact on the environment, travel time or length etc.

As the expression for the total sum is:

Total Sum = Cost + Heuristic (1 + Heuristic Head)
the Cost and Heuristic must have the same dimension and the Heuristic Head (HH) must not have dimension. In order to achieve, in a simple way, the same dimension of the Cost and Heuristic the same set of evaluation functions should be used for evaluating both the Cost and the Heuristic. The evaluation functions use a parameter set for the evaluation but the content of the parameter set is slightly different when used for evaluating the Heuristic than if used for evaluating the Cost. The idea is that the parameters of the Cost set shall reflect the past up to the current position and the parameters of the Heuristic set shall reflect the future from the current position to the destination. The Cost set is based on static and dynamic information for each segment of the path while the Heuristic set is based on the Cost sets for the path found so far, that is, they reflect some weighted mean of the path.

The search trees of a planned trip from boom_town to big_city with the conditions of figures 1 and 6 is found in figures 2 and 8 respectively. Noteworthy is that a longer trip was found for the area situation but the area was not avoided all together.

Figure 8. A search tree for a trip between boom_town and big_city as illustrated in figure 7.

References


Chapter 5

Directed Hierarchical Route Planning with Dynamic Information

Carl Brown and Per Lindevall
Directed Hierarchical Route Planning with Dynamic Information

Carl Brown and Per Lindevall
Knowledge Based Systems Laboratory
Swedish Institute of Computer Science
Box 1263, S-164 28 KISTA, SWEDEN
Phone: +46/8-752 1500, Fax +46/8-751 7230
e-mail: carl@sicse, perl@sicse.se

Abstract

One of the problems facing IVHS is the need to react to new information about the traffic situation in the area of the planned trip. Such situations may include information about traffic flow or interruptions, fleet management, traffic management for environmental or social purposes, or weather information affecting safety or trip time.

We show that efficient route planning can be done on the basis of a hierarchical map representation and a heuristic algorithm that takes advantage of the hierarchy. We then show that dynamic information can be incorporated in the algorithm using k-factors related to traffic flow, weather, or other events. Criteria for traffic management and pollution minimisation are expressed as the relationship between cost of traversal of segment and k-factors. Dynamic information is expressed as a new k-factor for an area-set of route segments (conceptually similar to a geographical area).

Key words: Route Planning and Optimisation, Navigation and Route Guidance.

1. Introduction

In this paper a map representation and a planning algorithm for a Dynamic RePlanning Car Navigation System are presented and discussed. A novel feature for this system is that it will take dynamic factors such as traffic flow and intensity or weather conditions into consideration when planning and monitoring routes. The task of car navigation is investigated purely from the planning point of view and, in this case, in a dynamically changing world. The questions tackled are: what is the nature of the dynamic information, what information is needed and how can the car navigator use this information? First we briefly present the problem domain in terms of our representation and algorithm. Second, plan generation through initial planning and plan refinement is described. In section three refinement with respect to incomplete and uncertain information, is considered. Finally some results are presented.

2. The Problem Domain: Map Representation and Algorithm

This section describes the map representation and the algorithm used to search for an optimal path in that map representation.
2.1 The Hierarchical Map Representation:
A hierarchical map representation [Bro90] is used in which all levels of representation are isomorphic. Any map object (i.e., town, region, intersection) may be described using a very limited set of primitive elements: expandable nodes, intersection nodes and segments. This makes our route-finding algorithm independent of the number of levels in the map representation. In fact, the number of hierarchical levels we use in the representation is not fixed.

The hierarchy is based on the assumption that main roads with a (presumably) higher speed limit and higher degree of maintenance are both shorter, safer and faster than regional roads under normal circumstances. (Route segments are given costs that reflect, for example, the distance covered or the time of travel.) This assumption holds unless a regional road could be a short cut or a main road is obstructed for some reason. This has proved to be a reasonable assumption about regional roads: the algorithm finds good routes under the assumption. As for blocked or restricted main roads, this is a case for dynamic replanning (see section 2.3 under Cost and Dynamic Factors and section 3).

To mirror this assumption, suitable groups of regional roads are clustered into an expandable node, representing a whole region of roads as a single expandable node. At a regional level, new expandable nodes can be encountered and the procedure is repeated. The same hierarchical treatment applies to the representation of cities.

Route finding means finding a path through the hierarchical representation. Once an expandable node is included in the path, it must be "opened up" to see if the weight/cost estimate is correct and to find the details of the correct path to traverse it. Figure 1 is an illustration of the map hierarchies.

![Hierarchical map representation](image)

**Figure 1 - Hierarchical map representation**

![Traffic flow curves](image)

**Figure 2 - Traffic flow curves**

For algorithmic reasons, we require that segments from an expandable node must lead to a non-expandable node. This means that two expandable nodes cannot be connected by a segment directly but must have at least one non-expandable node between them.

The smallest entity of the hierarchy is the intersection node. The intersection node is non-expandable and can appear at any hierarchical level. A final search path is found when all the nodes in the path are intersection nodes. All weight/cost estimates in the expandable nodes have then been replaced by actual costs.

The dynamic characteristics of a segment are based on statistical traffic flow information as described in [Gus89]. With knowledge of the conditions under which the information was gathered, weight-factors can be assigned to traffic flow curves. Each flow curve, and its associated weight-factor, will then describe the traffic flow as a function of time-of-day-and-year for the expected
condition. With knowledge of the expected traffic flow the travel time for the segment can be found.

2.2 Dynamic Information Represented as Areas:
In order to enhance route-finding with dynamic parameters such as weather and traffic congestions we have to superimpose this information on the hierarchical map structure. A natural way to superimpose these parameters (such as the effect of weather) is to define dynamic "areas" corresponding to the various, parameterized situations. If a chosen route intersects such an area, the corresponding cost function is updated (and possibly replanning has to be considered). By incorporating the travel time in the cost of traversing segments, the effects of the parameterized situation will be considered during planning for a route. By finding the optimum travel time during planning the worst-affected segments will be avoided by our route-finding algorithm.

Areas can be used for representing phenomenon such as polluted areas, traffic congestions and even scenic value. By associating some areas with unfavourable weight-factors (parameters), given for instance by the traffic information centres, we can make our route-finding algorithm avoid these areas at planning time. Areas and their weight-factors carry information on what flow curve the route-finding algorithm shall use when evaluating travel time for the considered segment.

![Figure 3](image3.png)
**Figure 3**
Dynamic parameterized area representation

![Figure 4](image4.png)
**Figure 4**
Prediction covering top hierarchical level

For traffic control purposes, areas can be assigned unfavourable weight-factors to suppress the use of these areas. The traffic information centre must then have knowledge about what weight-factor to use to get the expected behaviour from the traffic. Only cars that use a form of dynamic parameters for costs in their route optimization's strategy will be affected by messages from the traffic information centre. Area avoidance can be built into the heuristics such that messages from the traffic information centre always will be considered even if the planned route is an optimum on static parameters.

Air pollution is formed and spread in a manner similar to weather. In fact it is weather-dependent. That is, the area is a natural representation for phenomena such as pollution. Traffic congestion on the other hand is due to changes, expected or unexpected, to a single segment or an intersection between two segments. (Road work, two lane traffic yield into one lane, speed limit changes, high traffic density, accidents ...) The point here is that any of the mentioned causes are traffic density dependent and as such, when "critical mass" have been reached, will affect not only the segment with the actual road work but rather all adjoining segments to varying degrees. This means that a whole group of connected segments, spread over an area, will be congested due to the road work and the concept of area representation is applicable.

3
2.3 Heuristic Search

When searching a graph for a path, one is concerned with two different measures of the current path in order to be able to judge its expected performance against other paths that are candidates for the trip. These two measures can be described as "what has happened in the past" (i.e. what has been the cost of the route so far) and "what can be expected in the future" (i.e. what will the cost of the rest of the route be). The total cost of a path is simply the summation of the past cost and the future cost.

\[
\text{TOTAL\_COST} = \text{PAST} + \text{FUTURE}
\]

where \text{PAST} is the actual cost of the route planned up to the current point and \text{FUTURE} is the cost of the rest of the route. The measure for the \text{PAST} is known and is simply a summation of the encountered costs of the candidate path up to the current position. A measure of the future cost must be made. The future cost can be measured exactly by making an exhaustive search of all possible future routes and evaluating their exact costs. This is computationally expensive as the number of such possible routes is related exponentially to the number of possible path segments.

Instead, a heuristic search is made. That is, the \text{FUTURE} cost of a path is estimated using a heuristic method which tends to take the route in "the right direction". For example, a simple heuristic measure for the future cost of the path is based on straight line distance (possibly weighted) from the current position to the destination. Our heuristic search proceeds by making an initial plan that does not expand nodes in the hierarchy (an overview of the route) and then proceeds to refine the details of the initial plan.

The Initial Plan: We have taken this basic strategy, called the A* strategy, and adapted it for our hierarchical map representation. This provides a level of initial plan generation. We find an initial path that moves from the starting node, at level N of the hierarchy, to a final node at the same level or higher. The initial path never drops down a level in the hierarchy. It may traverse the graph at the same level, or climb a level as nodes are added to the path. The final node of the initial path is thus at level N or higher, regardless of the level of the destination or goal node.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>IF the current node of the chosen path is the goal THEN terminate the search.</td>
</tr>
<tr>
<td>2.</td>
<td>ELSE (Consider all successor segments to the current node:</td>
</tr>
<tr>
<td>2.1</td>
<td>Find all successors to the current node AND</td>
</tr>
<tr>
<td>2.2</td>
<td>Add the current node to the set of passed nodes.</td>
</tr>
<tr>
<td>2.3</td>
<td>Discard those successors which are members of the set of passed nodes.</td>
</tr>
<tr>
<td>2.4</td>
<td>Construct new possible paths, with the successors as their current nodes. Note: A successor segment can lead out of the current hierarchical level to a level above the current level.</td>
</tr>
<tr>
<td>2.5</td>
<td>Add the new paths to the set of possible paths.</td>
</tr>
<tr>
<td>2.6</td>
<td>Sort the set of possible paths on best total cost estimate.</td>
</tr>
<tr>
<td></td>
<td>Pick the possible path with the lowest total cost estimate as the new chosen path and repeat from step 1.)</td>
</tr>
</tbody>
</table>

All of the planning at level N (or higher) is completed without finding the details of the paths through the expandable nodes. After a path has been found and chosen, the expandable nodes in that path are examined in detail. To open expandable nodes may be viewed as plan refinement. That is, the details of the plan at lower hierarchical level are investigated.

Plan Refinement: The initial plan will contain intersection nodes and expandable nodes. Step 3 and 4 take an already chosen plan to investigate it with respect to expandable nodes. If any expandable nodes are found planning the path at the lower levels of the hierarchy is done as in steps 1 and 2. (We would like to point out that this may be done "when necessary", and need not be
done immediately after planning at level N only. Thus, possibilities for lazy plan refinement exist and can be exploited on route.)

Step 3. At the current level, N, traverse the list of nodes in the plan.
   3.1 IF the current node in the plan is an intersection
       THEN add the current node to the plan
   3.2 ELSE (if the current node is expandable)
       Find the previous and next nodes in the plan
       AND
       Search (steps 1 and 2 above) at level N-1 with the previous node
       as the starting node and the next node as the destination node. This
       sub-plan is inserted in the plan and step 3 continued.

Cost and Dynamic Factors: To cope with dynamic parameters we introduce a strategy for finding optimal routes, which uses an impedance factor [Gus90]. Impedance is associated with dynamic areas that the path will traverse on its way to the destination. Areas of easy access have a low impedance while areas of high impedance have a higher impedance. A path that considers crossing an unfavourable area will receive a higher total cost than the same path would have received if the area was not considered. The evaluation function can be explained as a terrain navigation or obstacle avoidance procedure. The expression for the cost evaluation function that considers impedance of dynamic parameters is:

\[
\text{TOTAL\_COST} = \text{PAST} \times \text{FUTURE}(1 + \text{IMPEDANCE})
\]

where IMPEDANCE is based on the amount of overlap, area content and the weight-factor for the area. Past and the future costs are evaluation is described next. Impedance is described in 2.3.2.

2.3.1 Cost Evaluation Functions for Paths:
There are two basic criteria for optimization of a route: the shortest-route and the fastest-route. If the shortest-route is taken to be a distance measure and the fastest-route to be a time measure, then the following distinction can be made:

- The shortest-route is a route found through search based on static information.
- The fastest-route is a route found through search based on dynamic information.

Each route found in a search will be optimal on one of the above strategies and sub-optimal on the other. Static information is data that does not change over the day on a regular basis. A change that can occur is that a segment has, for some reason, been closed while its normal state is open. Examples of static information are distance and road-tolls. Dynamic information is data that can be expected to change at any time. Typical dynamic data is time-of-traversal, average-speed or speed-limits (which may be controlled by road-traffic authorities during the day) and weather.

In the remainder of this section we develop the needs of the evaluation functions, both the PAST and the FUTURE, with respect to each criteria starting with the shortest-route.

The shortest-route, with respect to distance, can easily be determined by considering, for the Past function, the sum of actual segment distances, $d_{ij}$, from the start node to the current node and for the Future, the straight line distance, $d_{i}$, from the current node to the goal node. To enhance realism the straight line distance is multiplied with a distance factor, $f_d$, taken either from the previously considered segment or as some weighted mean of the encountered segments on the chosen route. The distance factor, $f_d$, is the quotient of the real segment distance and the straight line distance between start and end node of the segment.

The fastest-route, with respect to time, can be determined with slightly more work. What is needed here is the expected travel time, as described in [Gus89], for each traversed segment. By knowing the expected mean speed, $f_v(t)$, and the distance, $d_{ij}$, of the traversed segments a travel time can be found. For the Past function then, it is simply the sum of segment travel times from the start node to the current node. The Future are looking for an expected travel time that can
be calculated from the distance estimate of shortest-route heuristic divided by the mean speed, \( F_p(t) \), of the Past function. Thus the expression for the time estimate is: 
\[ t_h(t) = d_h / F_p(t) \]
The basic evaluation functions are given as:

PAST Function
\[ d_p = \bullet d^{l+1}_i \]

FUTURE Function
\[ \langle f_d \rangle \times d_h \]

Fastest-route:
\[ t_p(t) = \bullet \Delta t^{l+1}_i(t_i) \]
\[ \Delta t^{l+1}_i(t) = d^{l+1}_i / F^k_i(t) \]
\[ F^k_i(t) = \min(F^{l+1}_0(t), C^k_i(R)/R^k_i(t)) \]

Other evaluation functions have been developed including: Toll-route, Scenic-route, Fuel consumption and pollutant production.

2.3.2 The Impedance

The impedance is a factor based on the amount of overlap a straight line makes over the predicted area from the current planning position to the destination. The overlap to straight-line ratio weighted with the weight-factor for the area is the impedance for the area. In multiple weight-factor predictions the impedances is simply the summation of the part impedances. With several predictions the impedance is the sum of the impedances of each prediction. Formally it is described by:

\[
\text{IMPEDEANCE} = \bullet \left[ \bullet_j (\text{overlap}_{i,j}/\text{straightline})(k_{\text{default}}/k_{i,j}) \right]
\]

where

- \( i,j \) is area \( j \) of prediction \( i \)
- \( \bullet_j (\text{overlap}_{i,j}/\text{straightline}) \leq 1 \)
- \( \bullet_j (\text{overlap}_{i,j}/\text{straightline}) \leq 1 \) at a fixed \( i \)
- \( k_{\text{default}}/k_{i,j} \geq 1 \)

The impedance can be regarded as a penalty for trying to pass over an unfavourable area. The penalty is zero or more depending on the nature of the prediction (the set of areas). To determine the overlap a modified version of the Liang-Barsky line-clipping algorithm [Hea86] is used. Briefly, the algorithm is a method for summing the overlapping weighted areas in order to get a total impedance for a complex prediction (set of associated areas).

3. Criteria for Dynamic RePlanning

Dynamic replanning of routes is concerned with predictions and plan refinement. The interplay between the three can be complex. In this section we examine the possible effects. We believe, after having developed a conceptual architecture for a complete dynamic replanning navigation system, that the implementation of such a system requires careful attention to the needs of the driver. The following discussion will illustrate this.

3.1 Predictions

Our area representation is a way to guide and restrict the search algorithm in finding routes to the drivers liking. Using areas, the search algorithm will work in an obstacle-avoidance fashion. This is accomplished by letting heuristics "feel" or "sense" the future and avoid certain defined areas. The same area representation is used for mapping the dynamic information onto the static route map. In this way the same area will be used when evaluating the PAST function as well. An area is a restriction and as such also a prediction of what the future will look like [Lin91].
Each prediction is a sequence of snapshots in time with each snapshot having a defined start time and possibly a duration. The duration is not important as the start time of succeeding snapshots terminate the previous ones. This allows us to rely on the latest information we have in absence of anything better. Predictions also have an update cycle whose length depends upon the prediction type. If the update cycle time is shorter than the average predictions lifetime we have a situation where we have three cases to consider, (i) old predictions are confirmed on their validity, (ii) previously considered predictions are slightly changed and the path must be analysed with respect to this change, (iii) entirely new predictions show up on the planning scene and their impact on the path must evaluated. This analysis is relevant only if the prediction have been selected as "relevant" for the planning situation at hand.

3.2 Plan Refinement

The hierarchical map representation with its three map primitives suggested in [Bro90] has two major accomplishments. First a significant speed-up of more than 3 times over a flat map representation was achieved. Second the expandable nodes provide the means for hierarchical plans that easily refine to "executable" route guidance instructions. The expandable nodes can also be looked upon as a planning horizon, refining the plan by opening up the next expandable node in the plan as progression is made (the plan is executed). An exhausted plan, i.e. a plan that cannot be further refined, contain intersection nodes only.

A general procedure to initialize a plan consists of five steps: (i) find a path based on static information from Start to Destination, (ii) determine the "coverage" of the path, (iii) request dynamic information that can have an effect on the path, i.e. predictions which overlap the "coverage" from the previous step, (iv) analyze the path with respect to the new information. (v) update and install the new plan together with all relevant information.

The execution phase consists mainly of monitoring the progression of the installed plan and incoming predictions. Progression according to the plan (in time and space) is monitored mainly through dead reckoning and position matching against intersection positions. Monitoring predictions is necessary in order to avoid analysing the installed plan against irrelevant predictions and to still be able to use the best and latest information possible.

3.2.1 Incomplete Information

There are many aspects to incomplete information or incomplete knowledge about the "world". One aspect where incomplete knowledge is meaningful is in a situation where a strategy is established. A strategy is a plan that explicitly shows the intentions of the navigator, a plan for a future course of actions or an overview of possible courses of actions. For example, the initial plan can be regarded as a strategy in the respect that large parts of the plan (the expandable nodes) are not planned in any detail. What is known from the initial plan is that it is possible to reach the destination from the start position but exactly how this is done is not known. In order to find out, the expandable nodes have to be opened, i.e. a path on the map level below the expandable node must be planned.

To plan for immediate actions is called tactics. A tactic is a general, precompiled idea of what to do when the situation calls for immediate action. To open an expandable node can be looked upon as a tactic. Planning a path on a lower hierarchical level, that is, opening an expandable node can be viewed as a tactic as the area covered by the node (used for planning) is much smaller than the area covered by the original path from start to goal. To reflect on new information may be tactical. To continue as if nothing had happened may be a good tactic in a situation where the information is incomplete. Generally one does not know if the information at hand is incomplete or not, therefore the best thing to do is to look upon the information as complete until evidence to the contrary appears. Such evidence could be that the travel time on the current segment deviates from the expected travel time in the plan.

3.2.2 Uncertain Information

There are basically two types of uncertain information. These are (i) travel times based on information from the rapidly changing traffic flow curves at certain times of day and (ii) information based on predictions late in the update cycle.
The first type of uncertainty is based on the fact that planned routes can be time sensitive, i.e. for some routes it is important that time constraints are meet otherwise a small delay on one segment may cause longer and longer delays on segments ahead in the plan. This possible behaviour can be foreseen if the route can be measured on time vulnerability. The way to do it is for each segment in the path investigate the slope of the travel time curve at the expected time of traversal with respect to time. Routes or segments that are insensitive to time constraints have a slope close to or equal to zero. Segments with positive slopes will cost more to traverse when late to the expected arrival time.

The second type of uncertain information is not as easily dealt with. If a route is planned on information from a weather prediction, for example, predicting a very unstable weather situation, it is very likely that the prediction will have changed so much at the actual execution time that it is not valid any more. What will happen then is a situation very similar to the situation when incomplete information made the the plan invalid. This effect may be minimized with frequently updated dynamic information and effective monitoring of the dynamic information for relevance.

4 Some Results

The following performance results are based on a small map database with four hierarchical levels and eighty intersections. We expect the fifth and sixth situation below to be the prevailing situations in the real world and that the search space is large.

Tests have been carried out to establish performance differences between our hierarchical algorithm and the A* algorithm (using a flat version of our map database). One hundred twenty eight different runs were constructed representing six different situations of operation of the hierarchical algorithm. The six different situations are: (1) Paths that has been found on one hierarchical level. (2) A path going down the hierarchy and up again. (3) A path from a lower hierarchical level to a higher. (4) A path from a higher hierarchical level to a lower. (5) A path going up through the hierarchy and down again. (6) Paths that go up and down or vice versa several times before the goal is reached.

![Diagram 1 - Time vs Search Space](image1)

![Diagram 2 - Performance relative Situations](image2)

The hierarchical algorithm has accomplished what we set out to do: to assist the search algorithm in getting an broad overview of the situation before it gets down to details. By providing an overview, the hierarchical algorithm can enhance efficiency of the search algorithm. Two notable conclusions can be drawn from diagrams 1 and 2. First: the hierarchical algorithm performs more uniformly than A*. We can say that the variation between the hierarchical algorithm runs is smaller than corresponding A* runs, that the hierarchical algorithm search space is smaller (the ratio varies from 2.2:1 to 10.5:1 in our tests), that the longer the path the better the hierarchical algorithm performs. It is only in short paths and in paths one only one level where the hierarchical algorithm could not be expected to do better than the A* algorithm.
Construction of a very large database for validation of the dynamic area models is currently under way.

Acknowledgements:
The route planning system is embedded in a Dynamic Replanning Navigation System under development in Prometheus. We have benefited greatly from discussions with our colleagues Rune Gustavsson, Annika Wærn and Kristina Höök, without whose insight and constructive criticism we might have been lost in the detail of the problem.

References:


Chapter 6

Presenting Route Guidance Information: Some thoughts about interface design

Annika Wærn
Presenting Route Guidance Information: Some thoughts about interface design

Annika Wärn
Knowledge Based Systems Laboratory
Swedish Institute of Computer Science
Box 1263, S-164 28 Kista, Sweden +46-8-752 15 00

Abstract
The paper describes a model for the presentation of route guidance information, that is able to incorporate both static and dynamic route guidance within the same general interface structure. The model distinguishes between two modes that are dependent on the driver's knowledge:

- "Expert Mode": The driver has enough local knowledge to understand which route the system wants him to take.
- "Novice Mode": The driver has no knowledge about the advised route.

During a route guidance session, the driver might change between the modes several times. Based upon this distinction, a conceptual model for interface design is presented and motivated. The paper discusses to what extent a presentation system must check for and correct driver misconceptions. Such checking requires the development of a formal model of the driver's knowledge about alternative routes.

1. Introduction

Peter is driving through London. One part of the route he knows quite well; he drives it every morning on his way to work. However, this being a Sunday, he is not sure if the roads he has learnt to expect to be crowded weekdays are crowded on Sundays too, so he would like some guidance to choose the route that is fastest at the moment. Also, there is one place where he is not sure how to drive; he wants to get from one main road to another, but he is not sure if they intersect directly or if he has to drive some smaller roads to make the change.

Paul is driving the same route the same day. Paul has never driven a car in London before, and need extensive guidance through each intersection. Most of the time he does not know where he is, but he recognizes Buckingham Palace as he passes it.

The information that Peter and Paul need to efficiently drive through London is fundamentally different. It differs not only in how much information they need, but also in type. Most of the time, Peter only needs to get an overall description, consisting of road names and names of bridges and places. Peter will not be helped by such information at all; he must be told to turn right or left (or proceed straight ahead) in just about every intersection. Also, Peter will not be satisfied with a route recommendation that is inconsistent with his everyday experience unless it is given an explanation. For example, if the system suggests a road that Peter never would choose weekdays since it usually is heavily congested, he will likely not accept the advice unless the system assures him that today that road is clear. At one point, Peter is almost as lost as Paul, though. That occurs when he needs guidance to change from one main road to another.

Most of the time, Peter and Paul represent two essentially different driver types, the "expert" and the "novice". Since the expert and the novice have fundamentally different reasons for using the system, a fruitful approach to route guidance is to design a system to distinguish between these two types of drivers. Although the two modes require fundamentally different interface functions, these modes are related to the driver and not primarily to the system. Bluntly put: if the driver knows what he is doing, he is an expert, if he does not, he is a novice. As we see from the Peter example, a driver might switch between the modes during
one and the same driving session. When Peter is unsure of the intersection between the two main routes, he is a novice, whereas he an expert the rest of the time.

We define the notion of expert and novice mode more distinctly: Assume that A and B are the start and end points of a route, or the start and end points of a part of a route.

- Novice mode: The driver is ignorant of the area or of the advised route, and does not know how to (best) get from A to B.
- Expert mode: The driver knows one or several alternative routes between A and B, one of them being the one preferred by the system.

During a driving session, the driver can switch between these modes several times.

There is very little possibility for the system to infer which of these modes the driver currently is in. This implies that the driver should indicate to the system which of the modes he is in, for example by indicating what kind of information he wants from the system.

Expert mode is substantially more difficult to design than novice mode. For guiding an expert, the system must have some notion of exactly how much of an expert he actually is. For guiding a novice, the system can assume that all information is unknown.

This paper deals with
- What information needs to be presented in the different modes, and how it preferably should be presented.
- What are the unanswered questions about route guidance presentation, with respect to these two modes.

1.1 The Route Guidance Task

Before we start to discuss the interface functions needed, it will enhance understanding to discuss the varying purpose of a route guidance system.

A route guidance system is a help system directed to one particular driving task: the task of getting from a starting point to a destination point. The starting point is usually the current position (except when planning in advance), and the destination is usually defined by the driver, although it might be defined by a control centre (as in fleet management) or by the passengers (as in taxi driving).

The expert and the novice have fundamentally different reasons to consult a route guidance system. The novice would use it because he needs external help of some kind: without external advice he would not accomplish the task at all. Whenever he consults the system, he will benefit from it. The expert, on the other hand, would accomplish the task without aid from the system. He will benefit from using the system only in two situations:
- The system advice will lead to a route choice the driver judges is better than the one he would have chosen without the aid of the system.
- The route choice is complex enough to make it worth the trouble consulting the system, even though the driver is capable of finding a route that is good enough.

In both cases, the improvement must be large enough to make it "worth bothering", that is, the inconvenience of using the system must not outweigh the benefit of use.

These differences will affect the design of the presentation interface, as discussed later.

1.2 Dialogue related modes of a route guidance system

Apart from the driver modes (novice or expert) we have a set of modes related to the information content in the dialogue between the driver and the system. We can distinguish three different modes in the communication between the system and the driver.
• Driver input mode: What is it that the driver wants from the system?
In this mode the system gathers the information needed to fulfil its task: the start and
destination points, the criteria to be used for choosing a route and any additional constraints.

• Information presentation mode: Presenting an overall picture of the route or part of a route.
In this mode, the system will present the chosen route to the driver "in advance", not
necessarily before he starts driving, but before he has to make any actions that depend on
knowing the route. Thus, the information presented in this mode is not dependent on being
presented in a certain location or at a certain time.

• Instruction presentation mode: Route presentation for immediate action.
This is when the driver actually carries out the route description, by driving the suggested
route. Instruction presentation has time and/or location constraints.

It is important to remember that these modes might not always occur in this order, and that
the system might switch between the modes during a trip. The first and second mode require
much more active participation from the driver, why these modes might be difficult to
perform during driving. The interface design must for each mode take into account in what
situation the driver is going to use the system.

All of the dialogue modes can occur both in expert and novice mode; driver and dialogue
mode are independent axes in the design of a presentation system. But depending on whether
the driver is an expert or a novice, the input-output modes must be designed differently, and
the importance and complexity of each mode will vary.

Finally, it is worth pointing out that it is possible to conceive a fourth dialogue mode: an
explanation presentation mode. We have not considered this in the paper, since explanations
almost always are related to information given in one of the three modes defined above, and
we prefer viewing them as part of that mode. In section 3, we will explicitly use explanations
as a means for transferring information in the information presentation mode.

2. Novice Mode - traditional route guidance

We know quite a lot about the information needed when the driver is ignorant of the area.
This is the standard model for route guidance, and all commercially available systems so far
are of this type. I will here first mention what we know (or are reasonably certain) is true
about this kind of route guidance /AI 89/.

• The route guidance information needed is very short-range, local information. (You need
to know where to turn, which lane to be in etc.)
• The route guidance works best if the reference frame used is egocentric.
• There is little need for locality information or use of landmarks. A limited use of
landmarks visible to the driver can enhance performance by decreasing the risk that the driver
does an erroneous manoeuvre (as for example, turns right instead of left).
• There little need to explain why a certain route was preferred (since the driver isn't aware
of any alternatives). Still, the driver might request explanations, such as why a certain lane
was chosen, or, why we are heading in one direction when he believes that the destination is
in another direction.

Note that this knowledge only applies to the third dialogue mode of route guidance. However,
we can make some reasonable assumptions about driver input and information presentation.

As for driver input, the most crucial information we need from the driver is where to go. This
might be one or several goals, either specified by addresses, or by a map location. In the naive
user mode, it is extremely important that we can specify an address - the driver might not
know where to search for the address on a map. Also, it is worth noting that we might specify
goals by properties; for example, we might want to stop by a car wash, or at a gas station of a
certain brand, but exactly which one does not matter.
In some cases, it is important to get some knowledge about what criteria the driver has for a "good" route. In the novice mode, this is less important than in "expert" mode, as the driver's primary purpose for using the system is to be able to get from the starting point to the destination. If a system design for novice mode is based on the assumption that the fastest route choice also is the best according to the driver's wishes, this will probably not degrade performance of the system to the point where drivers stop using it.

In route guidance systems that are currently being (or about to be) marketed, the route presentation consists usually either of presenting the chosen route on a map, or giving directions as intersections are passed. The first presentation technique is more effective for the information presentation mode, whereas the second definitely is a way of instruction presentation. Another variant that exists, that can be viewed as a combination of information and instruction presentation, is systems that present the chosen route on a map, together with a marker indicating the car's current position.

The information presentation mode is of little importance in novice mode since it will provide the driver with little information he is able to use. A prestudy performed at SICS/Wå 89.90/ gave very little indication that this kind of information helped naive drivers. Further research is necessary to find out whether information presentation can be of any help at all, and if it can, what information should be provided to give useful help to novice drivers.

Several other topics remain to be investigated.

- How should the driver input mode be constructed to suit the driver's needs without being overly complex to use.

- What kind of landmarks and locality information will enhance performance, and to what extent should such information be volunteered by the system.

- What kind of explanations are wished for, in what dialogue modes, and to what extent should such information be volunteered by the system.

- Even though you can envision a design where route guidance information is continuously displayed, we cannot assume that that is optimal. For that reason, it remains to be determined when route guidance information is best presented, with respect to time, distance, driver attention and the current traffic situation.

- It is not at all clear how route information best should be presented: Visual or verbal presentation (or both); continuously or only sometimes; if it should be volunteered or given only on request.

3. Expertise mode - a new type of route guidance

Most drivers seldom drive in areas where they are completely ignorant. At least, drivers often know the main roads through an area. The driver knows most of the time at least one route from the current position to some position further along the route.

We have recognized two situations in which the expert still benefits from using the system. The first is when the route choice the consultation leads to is significantly better than the choice he would have made by himself. The second is when the route choice is complex enough to require careful consideration. We view these as the target situations for an expert mode route guidance system. The system must be designed to behave satisfactory in these two situations. This has implications as to what is required from the different dialogue modes.

3.1 The input mode

The driver input mode is mostly affected by the second target situation, as it must be designed to allow for complex requirements on the route. Examples are: The driver must be able to
specify several goals, or goals defined by properties. He should also be allowed to give
unusual constraints on route choice, or unusual criteria for "best" route. Furthermore, the
interface must allow for simple and fast input of these factors, since otherwise, it will take
less effort for the driver to figure a route out by himself instead of consulting the system.

3.2 The information presentation mode

As opposed to novice mode, information is most conveniently transferred to the driver in the
information presentation mode. Most of the information needed by the driver can be
presented beforehand; he will (most of the time) be able to understand a high-level
description of the route. As instructions, he will only need brief reminders, as well as some
kind of confirmation that he is "on track". On the other hand, he is likely to question or even
reject route choices during information presentation. Thus, the system will benefit from an
explanation facility.

Apart from requirements on the interface, expert mode also puts definite requirements on
what information the system must be able to access. If the system cannot access dynamic
information, a driver will use the system only to as long as he is learning from it. This effect
is well-known from the field of expert systems and exploited in intelligent tutoring (e.g. /Cl
87/). If the system behaviour is static, the user learns what answers to expect, and finally quits
using the system. The net effect is positive, since the user internalizes the knowledge
contained in the system, but fact remains: after some time the system becomes superfluous.
Thus, we view it as necessary that a system useful for route guidance for experts has access to
dynamically changing information, such as information about congestion, pollution, road
work and accidents, and that the system is able to use this information when choosing routes.

3.2.1 Is it actually necessary that the system presents a route?

In the discussions of this paper, we have so far tacitly assumed that a route guidance system
always presents a route to the driver. In expert mode, this is actually not quite necessary. If
we aim a route guidance system expert mode towards the first target situation, the case when
consulting the system leads to a better route choice, the system could aid the driver by simply
giving appropriate warnings. The system would then, when given the destination, decide on
what information and warnings might be relevant to the route, pass it on to the driver but
leave the route choice to him. If dynamic data changes during driving, the system could use
knowledge about where we currently are and where we are going to decide whether or not the
new information should be passed on to the driver.

This simple system design would give valid route guidance, but it will not address the second
target situation for expert drivers (complex route choice). One sample situation should really
be sufficient to convince the reader that a route suggestion is a good thing: assume that
relevant information changes during driving, requiring an immediate action by the driver, if
he decides to change his route depending on the received information. In this case, he will
benefit greatly if the system not only presents the information changes, but also (possibly on
demand) gives a route suggestion.

In the following discussions we assume a system that, at least on demand, presents a route to
the driver.

3.2.2 Information structures for expert mode route guidance

At SICS, we are currently planning a study concerning useful information structures for
expert route guidance /Hög 90/. We can assume that the preferred information will be of a
completely different type than in novice mode:

• The driver can make use of, and will probably prefer, long-range information.
• The reference frame might very well be local or global (as well as egocentric).
• The system can refer to landmarks outside the drivers view.
• Explanations might greatly increase the convenience and driver acceptance of the system.

Based on this, the information transferred to the driver in the information presentation mode for expert drivers should consist of a sequence of global, local (e.g. landmarks) or egocentric instructions, together with appropriate explanations about the route choice, such as significant data that was used when choosing the route. The presentation might use graphical, text or verbal output. Parts of the information, such as explanations, might be provided only on demand. We have two requirements:

• The description must be unambiguous.
• The description must be as concise as possible. (This holds also for information presented on demand.)

At first it might seem that the requirement that the presentation is concise only concerns the convenience of the system. But in order to get the system to work correctly, the driver must feel at ease with the system. Thus, it might sometimes be necessary to sacrifice the requirement that the description is unambiguous in order to achieve brevity. For example, egocentric information should be avoided as much as possible, as it requires lots of subsequent instructions and a high degree of attention from the driver.¹

3.2.3 Agreeing on a route description

If we assume that the system presents a route, the most important purpose of the information presentation is to present this route to the driver in a way that he understands. Unfortunately, it is not at all trivial to determine that a route description was unambiguous to the driver.

The obvious way to find a route description that is understood by the driver, is to present a description and then let the driver indicate that he has understood it. This negotiation dialogue could be very simplistic (as for example, a presentation produced by the system followed by a "understood-not understood" acknowledgement from the driver) or have a much more complex design (the driver might for example ask for specific clarifications or explanations). One factor that affects the dialogue design is whether it is performed beforehand, with the car standing still, or during driving. The dialogue format, as well as convenient means for presenting route descriptions of this kind, are important research topics.

Still, a system cannot assume that a route description really was unambiguous to the driver, even if he has acknowledged that he has understood it. Assume that the driver has been given as route description a sequence of instructions and a set of explanations. We can now have two sources of misunderstandings.

Firstly we have the human/cognitive factor: It might be that the driver has misunderstood the information, or do not use all of it to make his route choice. For example, we know from cognitive psychology /Ga 87 p. 89/ that there is a limited set of factors a person can consider at the same time (the figure 7 ± 2 is usually given in literature). Using this and similar knowledge about human cognitive processes when designing the route guidance format, we can make it less likely that the driver makes incorrect route choices because he has misunderstood or forgot information. Nevertheless, any interface design must be evaluated with respect to the likelihood of driver misunderstandings. This is one of the problems we expect to be investigating at SICS /HoWæ 90/.

Secondly, we have the background knowledge factor: Even if we assume that the information all correctly received and remembered by the driver, it is perfectly possible that he can imagine a route that fulfills both the sequence of instructions and the set of explanations, but

¹ Egocentric information is also inherently time and location dependent, which makes it difficult to use in the information mode. For example, an instruction like "turn left" must be given a time or location constraint to be unambiguous, such as "turn left after three intersections, by the oak tree", or following another instruction, "...and then turn left immediately".

6
still differs from the route chosen by the system. When the driver indicates that he has understood a route description, this only means that he has default route choices for all segments indicated by the instruction sequence in the description, that also fit the additional information given (as explanations). The route choices by the driver might differ from those of the system, because the driver might use information that differs from that which the system is using. Since we assume that the system is right (well, we are optimists, aren't we?), that implies that the driver is using incomplete or faulty knowledge. The system must be able to reason about the driver's knowledge.

3.2.4 What does the expert driver know, really?

To avoid errors due to incomplete background knowledge, the system should construct route descriptions so that they contain any information the driver does not know, but is dependent on to make a correct route choice. The information could either be given explicitly (e.g. information about a traffic accident on a certain road), or it could be implicit from the sequence of instructions (e.g. a sequence of landmarks is given that clearly indicates that a certain street should not be traversed). Ideally, the system should provide exactly what the driver does not know beforehand; no more, no less. Then what information can we assume the driver knows, and what information must be provided by the system?

Firstly, there is some evidence that people sometimes have conceptual errors in their mental model of areas and routes /Ga 87/. A person can erroneously assume one route to be shorter or faster than another one. But it seems likely that people usually have a fairly good understanding of these static properties of routes. If he has such misconceptions, they are probably not large. We can allow a system design to assume that the driver correctly takes into account static data, at least as long as nothing clearly points against this assumption.

It is much less likely that the driver is aware of dynamic data, but here we have a full spectrum of possibilities. The driver might very well be aware of some of the dynamic information. This depends upon, among other things, how long the dynamic data is valid. Some dynamic information such as information about traffic accidents, congestion areas and road conditions will change very quickly, and the driver can hardly ever be expected to know about them. Other data changes less quickly, such as road construction areas and temporary traffic regulations. I will subsequently call the second kind of data semi-dynamic. Some drivers, for example the driver that uses the system to get back and forth to work every day, will soon be very well aware of any semi-dynamic data governing his route. Drivers using the system while going through a well-known area which they visits less often might not be aware of dynamic data even though it has been valid for a rather long time.

A system design should not assume that semi-dynamic information is known to all drivers. A large group of users of a route guidance system are professional drivers, such as taxi drivers. These people have very good knowledge of static information, but will frequently have less knowledge of semi-dynamic information. Finding an optimal path is very important for these drivers. For non-professional drivers, it is not desirable that the system erroneously assumes that the driver has knowledge of semi-dynamic information, if an error leads to large detours. Conversely, the system should not assume that all semi-dynamic information is unknown either. Such a system would either present too much information to most drivers. We conclude that the optimal behaviour of the system changes with the driver's purpose. To illustrate the idea, view the following diagram.
It is possible to envision a whole set of design solutions. For example, the system can make different assumptions depending on how long a certain information has been valid, or we can allow the driver to choose in between different formats of the negotiation dialogue, or we can even allow the system to be personalized. Within the project, we should allow for experiments with several designs.

3.3 Presentation of a Replanned Route

During driving, the optimal route will change depending on different factors. The first and most obvious is that dynamic data might, and will, change during the trip. Secondly, the driver might depart from the optimal route. When is it necessary to inform the driver that the optimal route from where he is now to the destination has changed?

Here again, it becomes important to determine which route the driver is using as his default route\(^1\). Depending on how important it is to the driver that he is driving an optimal route, or how important it is that the system is non-obtrusive, several different choices can be made on when to present a changed route. Some possible choices are:

---

\(^1\) An interesting research topic is to investigate to what extent the driver's route choices during driving can be used to make better assumptions about which route he assumes as default.
• The replanned route is presented to the driver only if it was changed due to alterations of dynamic data.

• The replanned route is presented to the driver when it is expected to be "much faster" than what we assume is the driver's default route.

• The replanned route is presented to the driver when it does not fit the route description that has previously been presented by the system.

• The replanned route is presented to the driver even if it still fits the route description given, but does not coincide with what the system guess is the driver's default route.

In the original negotiation dialogue, the requirement was actually the last criteria; We required that the route description contained any information the driver needed to make a correct route choice. But this does not imply that the same requirements should be used once a description has been negotiated. For example, it might be difficult to get the driver to accept a replanned route, or to understand what has changed in the route description. Also, the presentation might have to take place when the driver is driving, which makes timing requirements harder and the possibilities for two-way dialogue very restricted. Here, we can only conclude that little is known from previous research, and that careful evaluation of any system design is needed.

4. Conclusions

We have defined the fundamental tasks for a route guidance system:

• For novice drivers: give help sufficient to get from the starting position to the destination.
• For expert drivers: provide a better route choice than the one the expert would have chosen, or aid the driver in a complex route choice situation.

We have also discussed how information best should be presented, depending on what descriptions the driver can understand. We claim that novice drivers will perform better if given egocentric instructions and have little need of beforehand information, whereas expert drivers can make use of beforehand information and have less need of timed instructions. These claims are partly supported by previous research in the area, however, most of the research so far has concerned novice drivers. For expert drivers, these questions need to be further investigated. In particular, there is a need for cognitive studies of the driver's understanding and acceptance of route descriptions.

We have, for expert route guidance, established the need for system knowledge of dynamic data. Finally, we have discussed in some depth what information the system needs to transfer to the expert driver. Ideally, the system should provide exactly what the driver does not know beforehand; no more, no less. We have concluded that there is a large set of data (semi-dynamic data) which will be known by some drivers and not by others. This uncertainty must be considered when a route guidance system is designed.

Finally, we have concluded that a route guidance system must be able to reason about the driver's knowledge of routes. For this reason, it is necessary to construct a formal model that allows reasoning about the driver's knowledge, that is powerful enough to handle the on-route situation where the driver departs from the expected route.
5. References

/Al 89/ Alm H., "Show me the way to go home - or drive me home, old chip". VTI report 1989.


Chapter 7

Route Guidance for Novice Navigators; Prestudy Results

Annika Wærn and Carl Brown
Route Guidance for Novice Navigators; 
Prestudy Results

Annika Wærn & Carl Brown
Swedish Institute of Computer Science, 
Knowledge Based Systems Laboratory

1 Introduction

We have performed a prestudy on route guidance. The study concerned drivers that are 
unfamiliar with the area. The topic was to gain experience that would help us to set up 
future, larger experiments.

We used a route where suburban, highway and in-city driving was mixed. The drivers 
were accompanied by a codriver who was familiar with the area of the route.

The goals were:

- To gather sample dialogues between driver and codriver as empirical data for use 
in developing a formal model of dialogue and an experimental dialogue 
management system.
- To get indications as to what information is normally used in a route description.

This report covers only some of the prestudy results. In particular, it does not contain 
any thorough analysis of dialogues. The dialogue analysis will be contained in a 
forthcoming SICS working paper.

2 Experiment setup

2.1 Experiment phases

The experiment consisted of three phases: first, the experiment subject was introduced 
to the experiment and shown a large-scale map picture of the route. Then we performed 
the trip, and finally we interviewed the subject while reviewing the videotape from the 
trip.

The trip was filmed using a normal video-camera filming through the windscreen. 
Sound was recorded on the same videotape by the microphone on the camera. One of 
the experiments sessions was carried out in a Volvo 240, the others in a SAAB 900.

We did not make any record whatsoever of the introduction. During the post-inspection 
terview we took notes, where we related comments to certain situations on the 
videotapes.

2.1 Unconstrained communication

In this experiment, we wanted to have the communication between the driver and the 
codriver as unconstrained as possible. The codriver was "expert" of the area, and 
allowed to use whatever information he/she thought appropriate to guide the driver. The 
driver was allowed to, and encouraged, to ask for any additional information he wished 
for. Throughout the experiments, we used the same, trained, codriver for providing 
information. The codriver was thus not an subject of the experiment.
The codriver had a set of notes describing the route to remember what to do during the trip. These notes may have restricted communication, both in form and content.

2.2 Route description

The route chosen consisted of several types of navigation situations; one part consisted of highway driving, another part consisted of following the main road through a suburban area, and a third part consisted of driving in a city setting where all roads were of roughly equal size and had similar traffic density.

2.3 Drivers

The drivers used were all fairly unaware of the area of the route. Some of them had a little knowledge of the area, and all recognized a few places along the route. The driving skills varied among the experiment subjects.

3 Prestudy observations

3.1 Classes of information

We found three basic types of information given or requested, that could be handled by a route guidance system.

- Orders: "Take left at the next red light", "Drive past McDonald's" etc. Orders were often repeated, often on driver request.
- Explanations: Why was this lane chosen, why do we have to do a U-turn...
- Non-route but map-related information: Where are we, what is the speed limit...

In addition to these information types, the driver and codriver often discussed the current traffic situation. Infrequently, they would actually refer to the traffic in route guidance.

Example 1 Subject no 2:

Codriver: ...And then we turn in on the other side... where the white car is turning.

On a couple of occasions, the codriver used knowledge about the route to guide the driver in tasks related to the traffic situation.

Example 2: Subject no 3:

Driver: Do I have stay in the left lane, or can I pass [the lorry]?  
Codriver: Well, yes [you can pass], but there is a road construction ahead, so it might be unwise to pass just now, since the left lane disappears.

These examples show that the route guidance subsystems as well as systems for more immediate driving tasks would benefit from interchanging information. In our experiments traffic information was never essential to the route guidance task, whereas (as in example 2) route guidance information was sometimes needed to perform immediate driving tasks correctly.
3.2 Timing

It is very difficult to get a picture about how far ahead the codriver gave directions. We only have actual times, and these are dependent on how slow/fast traffic is, red lights, obstructions etc. We tried to use these to estimate how far ahead the codriver gave directions, but failed to get a useful result. The reasons for this are further discussed in section 4.

We can make some statements about maximal warning times. It does not suffice to tell the driver that he should turn in the fourth intersection ahead; it is too difficult to determine what is an intersection and which intersection to start counting from. Also, information like "drive straight ahead for 2.5 km, then turn off" works only so as to tell the driver that he is free to change back and forth between lanes for a while. It does not prepare him for a turn.

It works to tell the driver to turn before he has the intersection in sight, but in this case he usually would want confirmation soon after the intersection comes in sight. This indicates a computational problem, as it is very difficult for the computer to determine when the driver has an intersection in sight. The driver's view can be obscured by other cars, the surroundings, building constructions, weather and light conditions etc.

3.3 Dialogue initiative

The dialogue was very free, and the codriver volunteered information most of the time. There were quite large differences to what extent drivers initiated dialogues. This could have been caused by the codriver behaving differently with different drivers, but we found no indication of this. Rather, it seemed to be personal differences. We can expect that in general, different drivers will want to initiate dialogues to different extent.

Drivers very seldom asked for explanations, even though such information was appreciated when it was given. In this experimental setup (with a human codriver and a driver that is unfamiliar with the area), the drivers are prone to trust the codriver's authority.

More commonly, drivers asked for location information. Interestingly enough, this did not happen when the driver was lost; The drivers asked for location information when they thought they recognised a place, and wanted to confirm this guess. We will discuss the drivers use of previous knowledge further in section 3.8.

3.4 Confirmation dialogues

We found that information was repeated quite often, and this kind of repetition was very often initiated by the driver. We have named this phenomena of driver-initiated repetitions confirmation. We can expect it to turn up in any form of computerized system, for example, the driver might look several times at a display.

Example 3 Subject number 2:

Codriver: ... in this intersection you turn right.
Driver: Turn right here.
Codriver: Yeah.

When the codriver has preinformed the driver about an intersection that is not in sight yet, there is almost always a request for confirmation. (In some cases the codriver repeated the information voluntarily.)

3
3.5 Reference frame of navigation instructions

In [2] three different frames of reference are mentioned:
• egocentric references, that use the location and and orientation of either the speaker or the guided person as point of reference,
• local references, that use to a named location or landmark as point of reference,
• global references, that refer to some global sense of direction (north, south for example).

In our experiments, we found two types of egocentric references: those that simply referred to the position or the orientation of the car, and those that also refer to some object in the immediate surroundings. Both are egocentric in the sense that they cannot be understood without knowing the position and orientation of the car, but the second kind also makes use of the driver’s external observations. We can call the second type of references "semi-local". These two types of references were by far the most common. Typical orders were worded like "Turn left at the next red light", using one egocentric and one "semi-local" reference. Lane changes were given using "left lane" or "right lane". The extensive use of egocentric and "semi-local" references implies that most instructions in novice route guidance have a very short "life-span", their correctness depend upon being issued at the right time.

Local references were used mainly for information, the route guidance did not depend much on local references. There was one exception: both the driver and the codriver could use the name of a road or a place if there was a sign with the name on it. This implies that the local references that occur in the dialogues actually are "semi-local": they implicitly refer to a object in the immediate surroundings of the car (the sign). For example, the codriver could give the command "Drive towards Roslagstull". This first reference was given with a sign in sight, but the driver then followed subsequent signs without the need for confirmation. This behaviour was mainly found on main roads and freeways. The driver would sometimes refer to a sign when asking for confirmation.

3.6 Landmarks noted by drivers

The term "landmark" is defined by K. Lynch [5] to denote any stationary object (tower, building, store, doorknob...) that the driver is familiar with. For the purpose of route guidance, we are more interested in places that the driver is familiar with. In Lynch’s terminology, these can be nodes, paths or locations of landmarks. Interestingly enough (and somewhat contrary to Lynch’s study), we found that different drivers used different
landmarks to describe the same place. For example: One of the intersections we passed was recognized by all the drivers. One of them mentioned the name of a café in the intersection: two drivers recognized the city library. One knew the name of one of the intersecting streets, and one recognised a certain store close to the intersection.

This points towards a specific problem: Even though it might be possible to find a set of places that is recognised by most people, it might be difficult to find landmarks that can be used as unambiguous references to these places.

3.7 Preinformation

In the experiments, we started out by giving the drivers a brief introduction to the route to be driven. This introduction was given by pointing out the route on a large scale map, and giving a set of "destinations" that described a fictive purpose of the trip.

All drivers remembered the names of the "goals" of the trip, but apart from this, only one of the drivers showed any ability whatsoever to use this information. This driver knew at one point, that we were supposed to turn left when coming from a small road onto a larger one. In this situation, he was able to relate his previous knowledge about the area to the information given beforehand, and this gave him enough knowledge to know in which direction he should turn.

3.8 The driver’s use of previous experience

The experiment was aimed towards guiding drivers unfamiliar with the area. Nevertheless, all the drivers had some previous experience of the area, either from driving through parts of it, or walking in some part of it. Several of the drivers made serious attempts to use this knowledge, sometimes successfully, sometimes not.

One example is a situation that occurred with subject number 1. At one point, we passed a street that the driver recognized, which lead to an area of the city he also was acquainted with. He tried to keep a notion about his direction with respect to the area he knew, although we were driving outside it. When arriving at the position indicated in figure 1, he assumed that we had arrived at a position in the area which he was acquainted with, partly because the areas had some similarities, but also because his sense of direction was almost correct. His estimate of the distance was poor, as he greatly underestimated the distance to the assumed position. (He had never driven in the area, only walked in it.)

---

1 Similar errors are reported in a study by Gärbling et. al [a].
The example shows that the driver's use of previous knowledge is a potential source of errors that should not be overlooked. If a system, on the other hand, could be made to understand what mechanisms a driver uses when he relates old experience to new information, it could actually help him to learn the area properly instead of just guiding him during a particular trip.

This problem, or opportunity, gives rise to a number of very interesting questions: How do novice navigators relate their previous experience to new situations? What kind of errors do they make, and what information is useful to make them understand the true situation? We believe these to be very difficult questions, that best are approached using experiences from cognitive studies in learning [4].

4 Evaluation of experiment setup and equipment

This section concerns the experiment setup as such, and does not discuss any results or conclusions drawn from the prestudy.

4.1 Interview phases

We tried to make the introduction as similar as possible for all subjects. All subjects were informed that they could ask anything they wanted, and that they were supposed to indicate when they were familiar with an area we were passing, or if they recognized some particular place or landmark. All subjects were also shown an overview of the trip on a large scale map, which gave them some idea about what areas were would pass through, and the essential structure of the trip. During the trip, the codriver sometimes referred to the information given beforehand.

In going over the results we decided that, since some useful information actually was given to the driver during this initial phase, it should have been recorded. For this simple prestudy, this was not a problem, but for any larger-scale experiment this is necessary. Else, it is difficult to find out whether drivers behave differently because they were introduced differently to the experiment, or if it is a personal difference.

The post-inspection interview should also have been recorded. The written interview protocols have two drawbacks: the references to the videotapes are not exact enough,
and the comments from the experiment subjects are not recorded in exact wordings. For the second point, a sound recording would do, whereas the first problem is solved only by filming the video display and recording the simultaneous discussion.

Apart from these technical comments, it is important to realize that interviewing people is a difficult art. We imposed no restrictions on the interviews, which allowed leading questions to be asked. In future studies, we must beforehand decide on what kind of situations we will want clarified, and how questions should be asked. If the interview is taped, it is also easier to discard answers afterwards, when we realize that a question or a discussion was too leading.

4.2 Videotape experience

The films taped within the car proved to have better quality than we originally expected. The view through the windshield is restricted, but anything that shows from the camera's position is clearly visible on the tapes. Road signs are clearly visible, most of the time it is possible to time intersection entries and exits, and it shows clearly when the car starts to turn or change lane. The only times when the visual quality becomes unacceptable is when the sun shines through the windshield.

The sound quality is also surprisingly good. Almost all driver and codriver utterances can be decoded, there are only a few exceptions. The background noise level is high, especially when the fan or windshield wipers are on. A high background noise level makes it very tiring to listen to the tapes, apart from making it more difficult to pick up the dialogue. For this reason, experiments should preferably be carried out when roads are dry, there is no rain, and the fan doesn't need to be used. (Since it is difficult to plan the weather beforehand, it is nice to know that the experiment works even if this is not fulfilled!)

Of course, we could not pick up much of the drivers actual actions using this experiment setup. Much to our surprise, it was possible most of the time to pick up when the driver put on the blinker, since this was heard on the tape. This can be used for timings, as long as one realizes that the fact that no blinker is heard doesn't mean that the blinker wasn't used. (Either, the blinker sound can be shadowed by other sounds, or the driver might just press down the blinker without pushing it to the position where it stays blinking.)

We also tried to use the videotapes for recording positions and timings. It was rather difficult to track the position of the car. Although it is at certain points possible to figure out a more or less exact position of the car (such as when entering an intersection), in between these points it is very difficult to get a good enough position from the videotape.

At the first glance, it might seem that it should be easier to time actions than to track positions. Moves, actions and utterances could be timed within a second by using the counter on the video camera, and better if using a separate stop watch. This might seem good enough, but for our purposes these timings proved to be almost worthless. We needed to time the time gap between forewarning and action; we wanted to check how far ahead of an action a driver was informed about what to do, and we wanted to check how far ahead of a turn or a lane change the driver put on the blinker. But the videotape timings gave no indication about this. Slow traffic or a red light would easily add half minutes to these time gaps. In one particular (but by no means extreme) intersection, all drivers were warned in about the same position, but the time between warning and turn varied between 21 to 68 seconds. Thus the true time gap is of little

---

1 At the time when these experiments were carried out, no results in this area were available from other research groups. Currently, several groups are doing valuable research in this area, and we will not put any effort in pursuing this research at SICS.
importance; what is important is when it is given with respect to when you expect to arrive at the intersection. For this kind of estimates you must know the position and speed of the vehicle when an action or utterance occurs.

In future experiments, we must be able to record position and speed of the vehicle. As long as the actions we are interested in are related to intersections within a fairly close range, we do not need the true position of the car, the distance to the intersection suffices. Since the entrance and exits of intersections can be timed, the relative distance would suffice to give reasonable estimates of how far in advance directions are given. For example: assume that the driver is driving at a speed of 50 km/h, and that the distance between the intersection and the point when the direction is issued is 100 m. Then we can estimate that the driver would have arrived at the intersection in seven seconds if not warned, and (assuming a reasonable braking force and a turning speed) it would take about four seconds before he needed to start braking if there was no traffic or red light restricting his speed.

4.3 Making protocols

We found it inconvenient to do a conventional written protocol from the videotape recordings. For one thing, utterances and actions might well happen in parallel. We wanted also to be able to record the position at which an instruction is issued. We also wanted to relate comments and explanations given in the post-inspection interview to the actions and dialogues during the trip. We would also like to be able to expand the information contained in the protocols during the analysis of experiments: We wanted, for example, to be able to record in the protocol if a traffic situation triggered a certain action or comment. Because of these wishes, we decided on designing a hypermedia tool for making protocols.

The tool was to incorporate the following properties:

- Any action or dialogue would be related to a position on a map.
- It should be possible to attach pictures cut from the videotape to an action or dialogue.
- It should be possible to attach comments and analyses to actions and dialogues.
- It should be possible to relate actions and dialogues to each other with respect to what action triggered what dialogue, or vice versa.
- It should be possible to get printouts of different parts of the protocol, as well as a picture relating the time overlaps between actions and dialogues.

A program incorporating most of these properties was constructed by Jöran Lindblom and used for making protocols for the prestudy. This proved to work fairly well, but using it has made us wish for a couple of additional functionalities.

• The protocol tool should allow us to define several different properties for a protocol fragment, and for example get a printout of all fragments that have a certain property, or fulfils a certain combination of properties. Currently there are only three properties: "action", "dialogue" and "external event", and these are mutually exclusive. During analysis, we want to be able to define new properties which might hold for the same dialogue fragment, such as "driver initiated" and "global reference".

• The tool should also allow us to attach comments that relate to a whole set of fragments, but not necessarily to a whole protocol.

The protocol tool we used had two technical drawbacks, the first and minor one was that we used a map with too large a scale. The second and more serious problem was that it proved tedious to include pictures in the protocols: the program for "grabbing" videotape pictures is difficult to use, and the quality of the picture that can be included in a protocol is poor. For this reason, we have not included pictures in the prestudy protocols.
We view this kind of software support for protocol development to be highly useful, although the current implementation needs to be refined if it is to be used in larger studies.

5 Conclusions

The purpose of this study was to find out what questions must be further studied, rather than drawing definite conclusions. The number of drivers was very small, which makes it difficult to draw general results from the material. Nevertheless, there are some conclusions that can be drawn from this study.

5.1 Prestudy results

Firstly, the study shows (as expected) that close-range, local directions work for aiding novice navigators. The drivers were fairly content with a codriver that mainly orders, as long as they were unfamiliar with the area. They did not demand explanations for the route choice, and very seldom requested location information.

We have also identified the confirmation phenomenon. From this study it is not possible to say whether confirmation is just a question of convenience, or if it actually improves performance of drivers. It would be very interesting to make a specialized study of the confirmation phenomenon.

One way of giving the driver access to confirmation is to have the instruction accessible continuously. It might for example be shown on a display as long as it is valid. Streeter et. al. [6] describes a study where recorded, verbal, orders were made available on request.

5.2 Research topics, novice route guidance

This prestudy has been mainly concerned with verbal instructions. Similar studies should of course be carried out where instructions are given visually, or by a combination of media.

Architecture suggestions for natural language interfaces in cars can be found in [1].

This prestudy points towards that finding a landmark to describe a known place might be a bigger problem than actually finding out what places people know of. Since landmarks will be used mostly for guiding expert drivers, this problem needs to be further studied in studies concerning expert navigators.

One important aspect of a route guidance system is how it can help a driver to get acquainted with an area. In this prestudy, we noted that drivers sometimes tried to relate their previous knowledge to the current trip, when they came to places they recognized. Further understanding of how drivers do this could be achieved in cognitive studies.

This study has given very little indication as to how far in advance drivers need instructions. It is important to do field studies on this, as it seems like very short forewarning times, somewhere in the range of 5 seconds, sometimes (depending on the speed of the vehicle) will do.\(^1\)

This prestudy indicates that there are two limits: one minimum-time limit which gives the last moment in which an instruction can be given, and one environment-dependent limit.

\(^1\) Some existing navigation systems (for example CARMINAT) uses the speed limit as a factor when deciding the minimum forewarning time. Evaluating studies of these systems is perhaps the most valuable way to gaining further knowledge about this.
limit which gives the earliest situation in which the instruction can be given in order to be remembered and carried out correctly.

Finally, the instruction format is dependent on where the instruction is given. For example, in natural language, the same turn command can be worded as:
"Turn right at the intersection after this one"
"Turn right in the next intersection"
"Turn right here"
Similar "format" differences occur with graphical displays.

The research group at SICS is currently focussing on studies of expert navigators [3, 7]; Drivers that are well enough acquainted with an area to find a route, although it might not necessarily be the best one.

6. References


Chapter 8

Some Principles for Route Descriptions Derived from Human Advisers

Kristina Höök and Jussi Karlgren
Some Principles for Route Descriptions
Derived from Human Advisers

Kristina Höök and Jussi Karlsgren
SICS, Knowledge Based System Laboratory, Box 1263, 164 28 Kista, Sweden
+46 8 752 15 00 (phone) +46 8 751 72 30 (fax)
kia@sics.se and jussi@dsv.su.se

March, 1991

Abstract

There is a need to make the interface of Route Guidance systems more flexible, so that they can adapt to the specific driver needs. Today's systems are primarily aimed at tourists, and interfaces for drivers that have more experience of a city have not been investigated. In this paper we describe a study with very experienced driver-navigators, where we have deduced principles as to how route descriptions are constructed and expressed by humans. Some of these principles are implementable, and a rough outline of a program is presented. Given a plan of how to go from A to B in a city, the program produces a verbal description of that plan. The goal is to incorporate verbal descriptions in Route Guidance systems, primarily aimed at driver-navigators with some knowledge of the city.

Furthermore, we speculate into what kind of cognitive processes are involved when humans choose and describe routes.

1. Introduction

In the field of Route Guidance, an important and difficult issue is how to describe a route, chosen by the planning part of the route guidance system, to the driver-navigator. We have identified the need of different descriptions for different kinds of drivers [Wärn, Alm]. As a first rough division we can differentiate between drivers who are unfamiliar with a city, like tourists, and drivers who live or work in a city. We name the first group tourist navigators, and the other group citizens navigators. Of course, there is no clear-cut line between these two groups, you might even belong to both groups in a city, for instance, if you know one part of the city very well, and another part not so well.

For the tourist navigator, it will be necessary to give very detailed route information, preferably during the trip, so that the driver never feels lost and knows exactly what to do in every intersection, as in the LISB system [von Tomkewitsch]. For the citizen navigator, this kind of instructions will be too detailed. Instead an overview of what route has been chosen, and a few helpful instructions while driving give enough guidance. The overview must, for safety reasons be given before the trip starts, as we do not want to interfere with the already complicated driving task. This puts high demands on the presentation form, length and content.

The starting point for the study reported here was a prestudy, reported in [Höök]. We asked 10 persons to describe a route first in verbal form, and then by drawing a map (the order was changed for some subjects). The subjects were encouraged to assume that the experimental leader did know enough about Stockholm to understand a simplified route description. This study showed that people are able to "chunk" a number of roads into one unit. For example, driving from the starting point to the nearest big road, E4, in our experiment, involved four roads. These four roads were "chunked" into the expression: "get down to E4". In the prestudy, we also found that the "chunks" obtained were different in verbal form as opposed to map form.
Apart from the interesting issue of importance of presentation form, we found the chunking mechanism interesting, since it could provide us with a means of reducing the length of a route description and still keep it unambiguous. Some questions were raised: when can something be turned into a chunk? what simplifications are allowed? does the city or the route need to have certain properties in order to lend itself to this kind of chunking?, etc.

In the study, we asked persons who have long experience of a city and who are from time to time asked to give route descriptions (taxi drivers, traffic policemen, etc.), to provide us with route descriptions. It must be emphasised that the study should foremost be looked upon as a data collecting study rather than an investigation into human behaviour. We are interested in finding results that can be implemented, and used as an interface to a route guidance system. In section 5, we shall make a rough outline of how we imagine that the results from the study can be used to make such an implementation. Even so, we have allowed ourselves to speculate upon what kind of mental processing is taking place when the subjects chose and describe routes, which is reported in section 4.

2. Experiment set-up

The experiment was conducted with routes in Stockholm, and the set-up consisted of 10 persons, out of which 5 were taxi-drivers, 2 were born and living in the city, and 3 were traffic policemen. One of the subjects was a woman. The analysis is based on eight of these ten subjects, two tapes were destroyed by the video machine. Note that some of the subjects knew the experimental leaders personally. They were paid SEK 75:-/hour, and the interviews took approximately 1 - 2 hours per subject. One of the subjects was English speaking, and the rest Swedish speaking, a deliberate choice since we wanted to see if there were any language dependent factors involved in route descriptions.

For each subject, we first asked them to describe their experience of Stockholm and whether they had been frequently exposed to describing routes to people and if so, if they thought that they were good at describing routes. The subjects were then given a pair of a starting point and a destination and were asked to do the following:

1. First find a route between the two points. Either ‘in the head’ or using a map provided by us. Subjects were encouraged to think aloud during this search.
2. Give a verbal route description of the chosen route to another citizen navigator (one of the authors). The map was taken away so that no “pointing” at the map would clutter up the results.
3. Give a verbal route description of the chosen route to a “tourist” driver (the other author).

(Throughout the rest of the document, we shall refer to the experimental leader who played the role of citizen simply as citizen, and the other experimental leader will consequentially be called the tourist.)

The order of 2 and 3 were varied and the subjects were allowed, but not encouraged, to chose different routes for the citizen and tourist. In some cases, we postponed either 2 or 3 until a few more pairs of starting points and destinations had been explored, in order to see whether the mere fact that the description were given right after one another would influence the descriptions themselves. The subjects were allowed to ask any questions they wanted about the knowledge held by any of the experimental leaders, or upon other issues, like whether the route should be optimal for rush-hours or not, if the route was to be legal or not, etc.

Finally, the subjects went through debriefing were we explained the purpose of the experiment. The whole session was video-taped.

We chose the six pairs of starting points and destinations to allow subjects variation in route choice and road size chosen: we wanted to investigate the conditions under which certain routes were preferred or differently described.

2.1 Stockholm as a Testing Ground

Choosing six pairs of points so that there would be alternative routes posed a problem: Stockholm is situated on the water, see figure 7 page 12, and geographical boundaries constrain free route choice in many directions. In addition, smaller roads in Stockholm often are
for one way traffic only, and many are reserved for local traffic use. The main traffic is required to follow major thoroughfares and avoid passing through residential areas, though restrictions are not always stringently adhered to. Many of the routes described to us violated the local traffic ordinances. Several of the subjects complained that the routes chosen were "impossible" and "too difficult", partly because the routes angled through the city on an East-West axis, and partly because we had chosen areas with extensive traffic planning restrictions.

2.2 Hierarchy

Using ideas from [Pailhouse, Streeter] we identified four hierarchical levels applicable in Stockholm. On the lowest level we placed small roads, used exclusively for residential and local traffic. On the next level we placed connecting roads. On the third level we placed roads with more than one lane, except highways. Finally, on the fourth and highest level, we placed highways and expressways.

![Figure 1. The hierarchy of roads used in the experiment.](image)

The starting and destination points where chosen so that routes that were likely to be chosen between the two points, would differ in terms of hierarchical level of the roads involved.

2.3 Patterns

We analysed routes chosen with respect to which "pattern" they adhered to. We had beforehand decided on three possible "patterns", A, B and C: shown schematically in figure 2. Pattern A is driving out of an area, driving on a bigger road for some time, and then driving into an area. Pattern B is driving out from area, driving on several bigger roads, possibly connecting these bigger roads via small roads. Pattern C is driving on small roads all the time.

![Figure 2. Patterns of routes.](image)

2.4 Distance

We chose the six start/destination pairs to be able to determine the effect of distance on the description. Two of the pairs have similar or identical starting points, and differ mainly in the distance to the destination.

3. Expectations

Why did we choose the properties of the starting points and destinations this way? One concern of ours, was that the results obtained from this study should be implementable. We were therefore foremost looking for that kind of properties, and the above mentioned all fulfil that criteria. Otherwise, we were inspired by the results from the prestudy [Höök], which clearly indicated that some underlying general principles could be extracted from human descriptions. The ideas of patterns, were also drawn from the work of [Elliott and Lesk, Streeter and Vitello], which clearly indicates that humans strive to chose routes so that they follow pattern A. [Pailhouse] has defined three levels of road nets and corresponding search strategies that driver in Paris make use of. From his study (and [Streeter]) we got the idea of a hierarchy of roads.
3.1 Chunks

From the prestudy [Höök], we had some ideas as to what a route chunk would be. For instance, the complicated issue of how to get onto a big road in pattern A, would be compiled into one chunk, and expressed as “get down to Strandvägen”. The patterns could be used as predictors of which chunks, and how many, a route description would consist of.

We expected a hierarchy of chunks, so that a description could be depicted and you could go from a very high level description with large chunks (each consisting of several roads), and then construct a new description where those large chunks had been broken it into smaller chunks, which in turn could be broken into even smaller ones, and so on.

We did not expect that chunking would be used with the tourist. That is, we expected that no roads would be skipped from the description.

3.2 Hierarchy

The hierarchical level of a road determines whether it will be mentioned or not. The higher the hierarchical level, the more likely it is that it will be mentioned to the citizen. Every road would always we mentioned in some form to the tourist.

3.3 Grammar

We were also hoping to explore the kind of concepts people use when they describe routes in order to get ideas of how to generate, automatically, verbal descriptions of routes. We expected to find a restricted language of route descriptions; a limited set of expressions with quite definite rules of how these expressions could be combined.

3.4 Route choice

If there are several possible routes between two points, we expected subjects to sometimes choose routes that are more easily described and/or more easily followed for the tourist.

We expected subjects to either chose different routes for goals along the same line but at different distances, or to change their route descriptions so that you would get approximately the same number of chunks irrespective of the fact that the descriptions should be the same for the first part of the routes that they had in common.¹

4. Results

As we shall see some of the anticipations could be confirmed but others had to be modified. We also found concepts that needed to be added to get a full description of how declarative descriptions work.

The influence of hierarchical level proved to be an important factor. We confirmed that the route descriptions follow a pattern usually consisting of driving from the starting point to some known big road, and then driving off the big road to the goal point. The subjects also chose different roads for the tourist and the citizen, sometimes explicitly stating that the reason was that the route described to the tourist was more easily described.

On the other hand, we were not correct in the assumption that the numbers of chunks mentioned would be kept the same for different routes along the same line but at different distances. We were also wrong in the idea of a hierarchy of chunked descriptions. Instead we found some really interesting properties of declarative descriptions, some of which explain when and why something is turned into a chunk.

We were also able to construct a grammar that together with some “heuristic” rules defines the route description language.

¹ The idea was drawn from the fact that people only keep 7±2 items [Miller] in their short term memory at one time. We expected the subjects to be limited to this, which of course, they were not due to the interview situation. Instead, it was the listener, i.e. the interviewer, who was limited to 7±2 items that (s)he could remember of the route descriptions given by the subject.
4.1 Three kinds of descriptions

When analysing the experiment we found three different kinds of descriptions, one more than we expected. Firstly, there is the tourist description which we call the *procedural* description for reasons we will explain below. There are two kinds of descriptions aimed at citizens, one which we could obtain just from taking the procedural description and crossing out some of the information, hereafter called the *mixed* description, and another based on an entirely different way of thinking about the routes, which we call the *declarative* description. In figure 3 we can see the difference between these three kinds of descriptions. (The example descriptions have been paraphrased according to the grammar in figure 5, for a full transcript of the example turn to appendix 1.)

**Tourist**

Drive Valhallavägen past two roundabouts up to a small “refug” that “stands out into the street a little” on Valhallavägen, where you turn left onto Artillerigatan. Drive it all the way down to Karlavägen where you turn right. Drive Karlavägen which is a boulevard past a park on your left-hand side up to the second left after the park where you turn left on Rådmansgatan. Drive it down past two red-lights, all the way until Sveavägen where you turn left. Drive immediately right onto Tegnérgatan. Drive up a hill to Tegnerlunden.

**Mixed**

Find your way up to Valhallavägen. Drive Valhallavägen up to Artillerigatan. Drive Artillerigatan down to Karlavägen where you turn left. Drive Karlavägen up to Rådmansgatan where you turn left. Drive Rådmansgatan down to Sveavägen where you turn left. Drive up to Tegnérgatan where you turn right. Drive up the hill and there is Tegnerlunden.

**Citizen**

Drive Karlavägen to Rådmansgatan. Drive Rådmansgatan to Sveavägen.

*Figure 3.* From Gustav Adolfs kyrka to Tegnerlunden in Stockholm; tourist, mixed and declarative descriptions.

The character of the procedural description aimed at the tourist (see also section 4.2) is foremost that it is a description of a *procedure*, namely, how to drive from A to B. Every road in the route is mentioned, and the subjects try to find properties of the road that the tourist can use for recognition. Subjects will even try to choose the route so that it contains as many recognizable features as possible.

The mixed description seemed to come up whenever the subject felt that the citizen interviewer did not know the route he was about to describe. The route could contain some new twitch not commonly used, or it could be that a part of the route was at a lower hierarchical level than the rest of the route. Usually, this kind of description was only used for a minor part of the whole route, but sometimes the entire route would be explained at this level. This kind of description is foremost characterized by the fact that it used ‘spatial markers’ like ‘left’, ‘right’, etc., and that we could obtain this description simply from crossing out certain extra information, like landmarks, lane information, etc., from the procedural description. Usually, no *objects* (roads, intersections or landmarks) were skipped from this description.

The declarative description, is most easily characterized as attempts to only mention enough road names to exclude any other possible route, or rather any other route that a citizen could possibly have chosen. It is a kind of reflection of the search space that the subjects traverses in order to decide upon a route. All the intersection and some roads have disappeared from this description, and there are not ‘spatial markers’ left.
4.2 What Does A Procedural description Look Like?

A procedural description can be seen as a sequence of identify-act instructions. Firstly, a point along the route is identified and then the action that should take place at that point is mentioned. Of course, the main place where actions happen while driving is in intersections. Thus intersections are central to procedural route descriptions. Actions that happen in intersections are to make a turn or to go straight through it, see examples 1 and 2 in figure 4 (the examples in figure 4 will be used throughout this section).

In procedural route descriptions we also found two other kinds of actions, namely to continue to drive along a road, and to pin-point a position. We call those the maintain and placement actions.

1. "Follow Karlbergsvägen until St Eriksgatan. Turn left there.
2. "Follow Odengatan until you get to the intersection Sveavägen/Odengatan. Continue on Odengatan."
3. "...you will get to a large roundabout: the Brommaplan roundabout. You will have a OK gas station to your right. And to the left you will have the Brommaplan bus depot. And you will come in and you want to turn left. Circle around the roundabout and make a left."
4. "...and to the thir... second roundabout after the Traneberg bridge. You want to make a left."
5. "And you go another two blocks up to the first traffic light. The first traffic light is across from a open area with a work of art you will recognize: a da Vinci thing."
6. "...until you get to a large intersection with a kiosk. To your left. On the other side of the intersection you will see a kiosk. Make a left there."
7. "...after you have passed a pretty big street called Engelbreksgatan - there's a pharmacy there, for instance - ..."
8. "And you will come out to big road, which you will make a right turn onto ..."
9. "Pulla Sveavägen all the way to a huge four-way intersection."
10. "You will get to a, what could one call it, a boulevard: like a very... it has two lanes in each direction, and a wide strip in the middle. With trees and parking and stuff."
11. "...as soon as possible make your way over to the left turn lane, to make a left turn..."
12. "You can continue along that road, past a large park on the left. All the way past that park. Then there's a slight kink in the road. Carry on on the same road. After a while that road will join another main road."
13. "Follow Roslagstunnel for some distance: say 500 meters. I would guess you pass six or seven intersections, and you will have to keep on the lookout for Ingmarstunnel..."
14. "First there is a left in to Sveavägen, and then immediately right from Sveavägen."
15. "Turn right in to Folkungagatan and you will go five or six blocks towards Medborgarplatsen."
16. "If you go north, you will come out pretty soon: you will have an incredible view of Stockholm. You are on Katarina hill. And you follow the slope you come out on there, which will slope down towards Slussen, with a view all along, on the edge of Södermalm."

Figure 4. Tourist route descriptions taken from the interviews (not paraphrased).

4.2.1 Identifying intersections

There are several methods of identifying an intersection. The most common way of identifying an intersection is by identifying the intersecting road (example 1 and 2 in figure 4). Another method is to identify it by name: "Brommaplan" (example 3), but very few intersections have names in themselves, and if they do, the names only very seldom are poster. Brommaplan, for instance, does not have its name posted.

The type of an intersection can be used for identification: "a roundabout" (examples 3 and 4). If the intersection is of an unusual type it is more likely to be used as a identifying feature: roundabouts, T-crossings, forks, and merges are all used in the interviews. Here we found the only result that indicates any difference between Swedish and English: in Swedish there are no names for forks and merges. Forks and merges are very noticeable when they occur along a route, and need to be mentioned. It took some effort for the interview subjects to clarify in Swedish exactly what they meant when they were describing the route to a tourist. The type of a intersection as a describing feature can only be done if no other similar intersections occur on the same route.

Intersections can also be referred to by using landmarks situated near or in the intersection, (examples 5 and 6). Landmarks have been defined by [Lynch]:

Landmarks are usually a rather simply defined physical object: building, sign, store, or mountain. Their use involves the singling out of one element from a host of possibilities. Some landmarks are distance ones, typically seen from many angles and distances, over the tops of smaller elements and used as radial reference. ... Other landmarks are primarily local, being visible only in restricted localities and from certain approaches.
The *size* of the intersection is sometimes used (example 6 and 9), but this is derived from the hierarchical level of the roads intersecting. An intersection is referred to as large when the intersecting road is at least as large as the street being traversed. One method that quite often turned out to be faulty, was to identify the intersection by the *number* of intersections before it (example 3, 4 and 14). Sometimes subjects would use several of these methods to identify the intersection (example 3 and 5-7). These methods can be (partially) ordered, so that name of intersection is most important, then comes type, landmark and size and thereafter intersecting road and number of intersections.

### 4.2.2 Identifying roads

Roads need to be identified as well. Roads are identified by *name* (any example) by *size* (example 8) or by *type* (example 10). The priority between the three is somewhat different from the identification of intersections. The name is very important. Even if the name of a road is new to a tourist it will be used. The fact that our subjects were allowed to use a map while planning the route may have influenced this [Gäredal et al.].

*Lane* information is very seldom used at all, but when it is it seems to take on two different functionalities. It is either a description of a road to help determine the size (example 10) or it is used to describe a turn in more detail (example 11). In our study relatively few routes make use of highways. Highways would probably generate more lane information.

### 4.2.3 Maintenance and placement actions

Beside informing about an action to be taken in an intersection, actions can be indications about how far to go to along a road. There are several different ways for indicating distance: number of blocks passed, distance in kilometres, or general expressions like “for a bit”, “all the way”, and so forth. These actions are not as much an instruction to do something, to initiate a new action, but rather to maintain a certain action, driving along a road, up to a choice point, an intersection (examples 12-15). They are needed whenever the road is going to be followed for an unusually short or long part.

The third type of instruction we find is a *placement* action: an instruction that explicitly anchors an abstract description to a location: a description of a place, an intersection, or a view. Placement instruction can be compositions of several landmark references. Placement actions tend to occur in the beginning of procedural descriptions, and then, later in the interview, when some characteristic object or view shows up, most often an object that is so visible that it absolutely must be mentioned (examples 3 and 15). In the latter situation the placement actions serve as a pause in the description. After the pause, the last part of the trip is often repeated.

### 4.2.4 Beginnings and ends

The unusual nature of the interview situation in which the descriptions are given (describing several routes, sitting in a room far from the places to be described, in front of a video camera) will influence several factors in it. A safe assumption is that the initial part of a route description definitely is expressed different than if given on the road. Several of the subjects suggested that if they had been at the spot, being confronted with a person asking for directions, they would have pointed in a direction to get the driver going.

In the interviews, tourists tended to be guided out from the starting point, and also got placements instructions for the starting point. The subject assuming that the tourist did not know what the starting point looks like.

Sometimes the end point would only be described by “then you follow the signs”, which is only possible when the end point is big enough to be posted. Otherwise the whole end part of the trip was described and a placement instruction was given so that the tourist would know when he had reached the end.

### 4.2.5 Defining a grammar for Observations

Bringing together what has been said above we find we have a small grammar for route descriptions, see figure 5. The strings it describes do not completely correspond to acceptable route descriptions, but our current goal is not to be able to generate good procedural descriptions, only to describe them as given by humans.

It turns out that most regularities of tourist route descriptions can be captured even with this relatively crude formalism. A small subset of natural language is involved. If we do not mind
the monotonous nature of the resulting dialogue, a simple template grammar will actually produce procedural descriptions. In a sense, the grammar strings are not a surface structure, but a deeper structure which simply contain a minimal amount of information to be realized in any way deemed fit. To some extent, it is language independent, one of the interviews was conducted in English with a native English spoken person, and the resulting material was completely compatible with the grammar.

| Route_descr → | Start Instruction* Goal |
| Action → | Goal → Placement |
| Look → | Goal → Sign |
| Action → | drive Road (RoadId) |
| (Until Intersection (IntersectionId )) | (where you Do) |
| (Direction) | (into Road (RoadId)) |
| Maintain → | drive Road (RoadId) |
| {past Landmark | through Area | Sign | towards NSWE | until namechange to Name | Distance} |
| Placement → | You can see | {Landmark | Road | ...} |
| RoadId → | which has Feature |
| Until → | until | all the way until | right to the end | ... |
| Intersection → | Name | Type | Size | an intersection |
| IntersectionId → | which has Feature | which is by Landmark |
| Do → | turn left | turn right | follow the traffic | go right through |
| Direction → | towards Area | towards NSWE | "down" | "up" | Sign |
| Distance → | N | meter | intersections | red_lights | blocks | ... | for a while | for a short while | directly |
| Sign → | following sign towards | {Area | Road ...} |
| N → | 1 | 2 | 1 | ... |

Figure 5. A descriptive grammar for tourist route descriptions.

4.3 Differences between tourist and citizen

Given that we know how a procedural description looks like, a mixed description can be obtained by deleting irrelevant information that the citizen navigator knows. The basic structure of this mixed description is not identify-act but rather identify as simple as possible and then act. By identify as simple as possible is understood that no hierarchical information, descriptions of landmarks, traffic, traffic regulations etc, are included in the description. What will be left is simply the backbone of a route description; "drive Road1 until Intersection1 turn left and drive Road2”. There are a few exceptions from this pattern. The road name can be substituted by a known citizen landmark, and subjects sometimes feel inclined to describe a road on a very low hierarchical level as "small". Otherwise, mixed descriptions can be obtained by removing the “IntersectionId” and the “RoadId” from the grammar rule “Action”, and take away the placement and maintain instructions from the grammar in figure 5.

As indicated above, declarative descriptions on the other hand, are fundamentally different from procedural descriptions. We cannot simply take a procedural description and cross out irrelevant information and obtain a declarative description.

| Route_descr → | Start Instruction* Goal |
| Instruction → | Citizen |
| Start → | [] |
| Goal → | [] | Sign | Mixed |
| Citizen → | drive Road | drive past Landmark |
| Sign → | following sign towards | {Area | Road ...} |
| Mixed → | see above. |

2 For an example of dialogue that is paraphrased using this grammar see the three example descriptions in the beginning of section 4.1.
Figure 6. A descriptive grammar for citizen route descriptions.

Following the concepts for the procedural description, we see that actions for citizens do not happen in intersections. Instead, it seems to be the case that we have a more declarative description, that does not explain how you get yourself through a number of intersections with left and right turns. The description will instead consists of a number of citizen instructions that in most of the cases is simply a road name, formalized in the grammar in figure 6. Sometimes the subjects will say “drive Karlavägen”, but sometimes, they even exclude the action of driving on the road and state the road by itself “Karlavägen” or “you take Karlavägen”, cf. figure 3 in section 4.1.

4.3.1 Gaps in the declarative descriptions

In the procedural descriptions, every road, every intersection where action should take place, and some of the landmarks were mentioned. In the declarative description, all the intersections, and almost all the landmarks disappear. It is even the case that some of the roads are left out. Which ones? The issue of which roads are left out is a complicated one. In table 1 we have counted all the times one or several roads were not mentioned in the descriptions. We have differentiated between when roads where left out in the beginning of the route, the middle of the route or the end of the route for each of the six roads. As you can see, tourists (almost) never get gaps in the middle of the trip. On a few occasions the procedural descriptions had gaps in the beginning or end of the trip, as for road2 where three subjects left out the last road or two.

<table>
<thead>
<tr>
<th>Road1</th>
<th>Road2</th>
<th>Road3</th>
<th>Road4</th>
<th>Road5</th>
<th>Road6</th>
</tr>
</thead>
<tbody>
<tr>
<td>B M E</td>
<td>B M E</td>
<td>B M E</td>
<td>B M E</td>
<td>B M E</td>
<td>B M E</td>
</tr>
</tbody>
</table>

Table 1. Gaps in the route descriptions.

For the citizens, many more roads disappear from the descriptions. Firstly, subjects seem to take away the first part of the trip. The subjects apparently assume, that the reason for asking about a route between the starting point and the destination, is that you know where you are, but you do not know exactly where the destination is situated. So usually, the first one or two roads are taken away. The principle seems to be that all roads before a road with quite a high hierarchical level, can be taken away, if the number of roads is sufficiently small (2-3 roads).

The last part of the trip is not described with the same pattern, instead the simplified declarative description is often used by the subjects. An exception seems to be when the goal is on a very high hierarchical level or close to a road which is, as we can see with the roads number 1 and 2 where eight respectively seven subjects left the last roads out. It can be observed that the hierarchical level of the starting point does not seem to affect whether it will be described or not, even when it is very low. See roads 2,3,5 and 6 for roads with low hierarchical level of starting point, where as many as 4,6,6 and 4 subjects respectively have left roads out in the declarative description.

As we see in table 1 there are also gaps in the middle of the description where roads have been left out, and the principle behind those gaps is much more complicated to depict. It seems like that the roads that are mentioned serves the purpose of excluding other possible routes. By other possible routes, we do not intend any possible route, but rather any route that would have been a likely candidate to get to the destination. This does not seem to be the whole explanation, it also seems like the subjects add some roads the make the route description complete, and that those roads are usually on a high hierarchical level. Any road which is only going to be followed for a minor part of its length, or that is on a lower hierarchical level than the rest of the roads, are only mentioned if they give the route an unexpected twitch or if they lead to the goal. The roads that are to be driven to their end are usually mentioned.
Subjects abandon this pattern when a 'strange' route has been chosen. Something seems to be a strange route whenever roads on a low hierarchical level are chosen, or when the chosen route is much longer than the crow-flight distance. The mixed kind of description will then be used.

As we see the notion of patterns can serve as a predictor to cover most of these irregularities. In a pattern A route, only beginnings and ends can disappear. In pattern B and C, we need to look out for places in the middle of the trip where roads can be taken away.

4.3.2 Landmarks in procedural and declarative descriptions

Subjects use landmarks in quite different ways for citizens as opposed to tourists. The obvious difference should be that some landmarks are only visible/usable to a driver if she knows the city. For instance, one of the subjects described a road to be one quarter away from a big intersection known as Roslagstull, where the driver at that point, would have been unable to see Roslagstull. Another subject talked about a portion of the road as Tre Portars backe (Three Gates Hill), which is a name that is derived from the fact that there used to be three bridges over that road, but nowadays the bridges are no longer there.

As seen in table 2, tourists in general were provided with a lot more landmarks than were citizens. In the table we have divided landmarks into six groups. The first group, Stockholm, is landmarks that are only comprehensible to a citizen of Stockholm. The second group, geographical landmarks, is mainly hills, parks and other similar features. The two groups that possibly should have been included in the geographical landmarks group are the surroundings and traffic landmarks. The surroundings group consists of description of views like "you will follow water to your left", or "then you will have a view office buildings all around you". The surroundings group differ from the geographical group in that it describes objects that in a sense are continous, while the geographical objects are points. The traffic landmarks on the other hand, are descriptions of the route in terms of traffic regulations, or the number of lanes, or suchlike. The building landmarks are the ones that are most typically landmarks in the Lynch sense. They are usually visible from the road. In this group we have also included hotels, pharmacies, bus stations, etc., that in some sense are buildings and recognizable to everybody. Finally, we have made bridges into their own group since they in a sense belongs to both the traffic, geographical and buildings groups and also are so frequently occuring in the descriptions due to the infrastructure of Stockholm.

<table>
<thead>
<tr>
<th></th>
<th>Sto</th>
<th>Geo</th>
<th>Surr</th>
<th>Tra</th>
<th>Bui</th>
<th>Bri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Novice</td>
<td>0</td>
<td>45</td>
<td>34</td>
<td>70</td>
<td>80</td>
<td>28</td>
</tr>
<tr>
<td>Expert</td>
<td>16</td>
<td>14</td>
<td>2</td>
<td>13</td>
<td>24</td>
<td>17</td>
</tr>
</tbody>
</table>

Sto = Stockholm, Geo = Geographical, Surr = Surroundings, Tra = Traffic, Bui = Buildings, Bri = Bridges

Table 2. The number of times landmarks were mentioned in the route descriptions.

With tourists we found that landmarks are used for three different purposes. Firstly, they make it possible for the tourist to recognize an intersection where an action is supposed to take place (example 6 in figure 4). Secondly, they can be used to keep the tourist on a certain road for a while, in the maintenance instructions (example 12). Thirdly, the placement instructions usually consists of one or more landmarks (examples 3 and 15), used in a more checkpoint manner, to indicate that a new series of instructions will follow after that checkpoint.

With citizen navigators on the other hand, landmarks are used rather as choice points instead of recognizable items. Instead of using a road name a landmark that uniquely determines that road can be used. The landmark is therefore not described in any detail, but is only referred to by name.

How often are landmarks used? For the tourist, it seems to be as often as possible. Whenever there is a noticeable landmark in an intersection, it will be mentioned. An intersection without landmarks, is described by some other means, like name, intersecting roads, hierarchical level, traffic intensity, or something else. Landmarks along the road, only appear when you have to travel for some longer time on that road, as in maintenance actions.
In the declarative descriptions, landmarks quite frequently turned up when the subject could not remember the name of the road, or when the landmark was such a well known item that its name would supersede the road name.

**4.3.3 Hierarchy of roads and intersections**

In our assumptions (section three) we assumed that hierarchical level of roads would play a role in the choice of roads and the way they are described. It turned out that our subjects were so aware of the hierarchical level of objects that they even would tell the tourist about the hierarchical level of roads or intersections in their descriptions.

The citizen was usually not told about the hierarchical level of roads or intersections, which is natural since the subjects assumed that they knew it from the start. The few times it occurred was whenever the hierarchical level of the road was unusually low. Otherwise the subject probably assumed that the citizen knew the hierarchical level of roads. In table 3 we have put together a list of how many times the hierarchical level was mentioned to the tourist and the citizen. The third use of hierarchical information, “past N intersections”, is when an intersection is identified by a phrase like “you driver past N small intersections until a big one”.

<table>
<thead>
<tr>
<th>Intersection described using hierarchical evaluation</th>
<th>Road described using hierarchical evaluation</th>
<th>Past N intersections</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Novice</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Expert</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Intersection = intersection described using hierarchical evaluation, Road = road described using hierarchical evaluation, Past N intersections = hierarchical information used to determine where an action of turning should take place.

**Table 3. Use of hierarchical information.**

**4.4 Route Choice**

One of the pairs of starting and end points did not allow different route choices, but the other ones did. They were chosen so that the possible routes between the two points would differ in terms of patterns, see figure 2 on page 3. We found that in quite a few cases, the subjects would chose different routes for the tourist and the citizen, see table 4. Usually, when they choose different routes they would mention the fact that they did so, and sometimes why. Usually the reason was that the route chosen for the tourist was more easily described.

<table>
<thead>
<tr>
<th>Road1</th>
<th>Road2</th>
<th>Road3</th>
<th>Road4</th>
<th>Road5</th>
<th>Road6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different routes</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 4. Numbers of times different routes for chosen for the citizen and the tourist.**

Now, what makes a route more easily described than another one? [Davis and Trobaugh] have defined some properties of easily described routes. They say that an easily described route should not contain too many turns, and it should, when possible, go via important landmarks. One of our subjects explicitly verified the second property. He consequentially tried to get the tourist past big landmarks that were visually attractive. The first hypothesis could also be verified. The tourist route was usually more straight ahead, included fewer turns.

We would like to add another property that has to do with hierarchical level of roads. Our subjects frequently tried to chose roads that were on the highest hierarchical level when possible. The concern seems both to be that the route should be the shortest in terms of time, but also that, especially tourists, should not get lost. A bigger road is easier to recognize, it has.

---

3 From one interview: “I am thinking about whether I should tak... try to explain the same road a gave Jussi, or if there is any easter one. It is not certain that ... It is not certain that the shortest route is the best to explain, right?”
better signs, and you can usually travel on it longer than on a road on a lower hierarchical level.4

In table 5 we have illustrated the difference between the tourist and citizen routes going from Katarina church to the university (road2 in table 4 above) in those cases where different routes where chosen by the subjects. The numbers refer to the hierarchy of road types described in section 2. You can notice that the number of roads appearing in the citizen route are slightly higher, but the interesting difference is that the hierarchical level is in general lower for the citizen.

<table>
<thead>
<tr>
<th>Subject#1: Expert Novice</th>
<th>#obj</th>
<th>#descr-obj</th>
<th>Hierarchical level of the roads involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert</td>
<td>12</td>
<td>4</td>
<td>4-3-2-3-3-2-2-2-2-2-2-1-4</td>
</tr>
<tr>
<td>Novice</td>
<td>10</td>
<td>10</td>
<td>4-3-2-1-1-1-2-3-1-4</td>
</tr>
<tr>
<td>Subject#2: Expert Novice</td>
<td>10</td>
<td>5</td>
<td>4-3-1-1-1-2-3-2-1-4</td>
</tr>
<tr>
<td>Subject#3: Expert Novice</td>
<td>9</td>
<td>8</td>
<td>4-3-2-2-3-2-2-1-4</td>
</tr>
<tr>
<td>Subject#4: Expert Novice</td>
<td>12</td>
<td>5</td>
<td>4-3-2-2-1-1-2-3-2-1-4</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10</td>
<td>4-3-2-2-1-1-2-3-2-1-4</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>8</td>
<td>4-3-2-3-2-2-2-2-2-2-2-2-2-3-1-4</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>8</td>
<td>4-3-2-2-1-2-2-3-1-4</td>
</tr>
</tbody>
</table>

#obj = number of roads in the chosen route, #descr-obj = number of described roads

Table 5. Differences in number of roads, number of described roads and hierarchical level of roads in tourist and citizen routes.

Different routes were also quite frequently chosen for goals that were along the same line, but where one goal was much further away than the other. An interesting pattern becomes apparent. When a route is chosen, it seems like humans first try to backtrack from the goal in the direction of the starting point, to a well-known spot or road to which they know that they can find their way from the starting point.

![Map](image)

Figure 7. Going from Karlaplan to S:t Görans hospital or Drottningholm.

Now, in figure 7, we see that the one of the goals that we chose for our study, lies very close to a big road, marked as Essingeleden. This big road was very frequently chosen for the second goal, Drottningholm, but not as often for the first goal, S:t Görans hospital, that lies so

4 On the other hand, one person rejected a bigger road for estethical reasons. He felt that a tourist should go on beautiful roads, and also that a big road has the disadvantage of being to hard to get off at the right place. If you miss an exit you might sometimes have to travel several kilometres before you can get off and turn back.
close to it. We found that since chosing this road would mean going in the wrong direction a couple of blocks in the direction from the starting point, subjects were unwilling to make this connection, both for tourists and for citizens. Instead they connected the point Fridhemsplan with the goal, and tried to find a route from the starting point to Fridhemsplan. The route through Fridhemsplan is much more complicated both to describe and to ride. Essingeleden is a route at the highest level of hierarchy, and the speed-limit is here 70 km/h instead of 50 km/h which is the normal speed limit in the city.

To some extent the way that the subjects construct the routes in their minds, will manifest itself in their description. In the example above, Fridhemsplan would be mentioned as an important point in the declarative description if that route were chosen.

Another pattern was found with the second pair of goals that were situated along the same road. The closest goal, Ingmarsgatan, was this time very close to the road that people normally used for the second goal, Stockholms University. Different roads were still chosen for the two goals, but the consideration did not seem to depend upon the fact that you would have to travel one block ‘back’ from Roslagstull, in order the reach the first goal from the chosen road. Instead the ability to describe the road in an as unambiguous manner as possible was the criteria of whether one route would be chosen before another. One route is much more difficult to describe since it involves more objects, and the objects are at a lower hierarchical level. As a consequence this route was more seldom chosen, and usually only chosen for the citizen.

4.5 Why these descriptions?

Why do people describe the routes as they do? The issue seems to be a mixture of how the route is chosen, which mainly manifests itself in the declarative description, and whom the subject thinks he is directing himself to.

If the subjects believes that he is directing himself to a tourist, you get the sense that the subject is mentally tracing the route, observing features as she goes along. The subject is constantly placing herself and the tourist ‘in’ the route with expressions like “then you will be at ...

“then you will see ...”, “then you go under a ...”. When doing the interviews, it was obvious that the subjects were trying hard to visualize the route in order to give as good placements descriptions as possible. One of the subjects even talked about the difficulty he experienced in visualizing certain parts of the route.5

In the study, we could see that when subjects try to visualize the route, the description not only tends to be much longer, it also takes much longer time to describe the route. It seems that visualization is a costly, cognitive process.

If the subject is directing herself to a citizen the descriptions seems to be much closer to how the subject has constructed the route in her mind without visualisation. You get the impression that she is viewing the city from above, seeing important roads and points where choices must be made.

5. Automatic generation of messages

We have described the study and the kind of principles we could extract from it. What can we do with these principles? We would like to do a ‘sensible’ implementation of these principles. Since ‘sensible’ is not necessarily equal to how humans describe routes [Riesbeck], we need also to look at how humans understand and misunderstand route descriptions. Luckily, there have been quite a number of such studies [Streeter and Vitello, Labiale, etc.]. We know that humans are bad at reading and understanding maps, bad at estimating distances and number of blocks, that humans tend to forget instructions with more than 7±2 items, etc.

5 “I am mostly trying to visualize the route myself at the moment. And see... And count the... The difficult points. The points that you could generally worry about: directions, difficult crossings. [...] OK. I think the difficulty is for me that I’d like to picture the route in front of me as I describe it. That’s what I have difficulty doing.”
Let us set up a scenario of how a route guidance interface could look like. Let us assume that we shall only be able to provide the most simplistic kind of user modeling, one where the user herself indicates which of two modes she wants to use, a kind of static user modeling.6

The first mode, the tourist mode, would not give any route information to the driver prior the trip, which is easily justified by the fact that the tourist does not know the city and would not be helped by getting information about things she cannot see yet. Instead very exact instruction would be provided during the trip, helping the tourist through each intersection. For instance, parts of the LIS8 interface could be a nice alternative for the during trip information, see [von Tomkewitsch]. The tourist mode would not contain any explanations as to why a certain route has been chosen. The tourist is simply interested in getting to the destination using a nice and easily followed route.

The second mode, the citizen mode, would provide somewhat a different approach. Firstly, it would give route information before leaving. This would help the driver to get a sense of where she is going. If the driver is very experienced, it could even be the only way the route is presented.

During the trip, we can imagine two different approaches. One would be to give the same kind of instruction that the tourist is provided with, which might be tedious for the citizen to listen to. The other approach would be to use something similar to the mixed description divided into manageable parts. Further investigations with an implemented system would provide more insight into what kind of solution might be best.

For the purpose of commuters, or very experienced drivers, we might even consider giving explanations before the trip starts as to why a certain route has been chosen. How to do that requires a lot of insight into how people reason about the mechanisms that control traffic intensity, and it might even require that the system knows which routes are likely to be chosen by humans and which are not. We believe that especially with commuters, it will be necessary to investigate this further, for a more lengthy discussion see [Lindevall and Höök].

In table 5 we have summarized our view. We could imagine a situation where the driver would switch on or off citizen mode, to obtain all three levels of information. If the system is operating in tourist mode, only on-road information will be obtained. If citizen mode is on, no during trip information will be provided. If both are on, both kinds of information will be given.

<table>
<thead>
<tr>
<th></th>
<th>Before</th>
<th>During</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tourist</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Middle</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Citizen</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 5. Route guidance for different drivers.

Let us explore the system for plan presentation. Given a starting point, a destination, knowledge of the user's experiences in the area in a user model, and a database of the route to be traversed the strategy for description will be based on a three step process, see figure 8.7 First the planner constructs a route. It consist of nodes and segments, where the nodes roughly corresponds to intersections, and the segments to roads. The hierarchical level of the segments is included in the plan.

---

6 We could imagine a situation where the driver identifies herself through a plastic card used as a key to the car, on which we have put a more dynamic user model that adapts to the user behaviour, habits and knowledge. There is a lot of research effort put into such dynamic user models, but for the moment it is probably more realistic to use a static user model.

7 In figure 8 we have left out a number of components necessary for a route guidance system, for instance, a route monitor, some integrated dialogue management, dynamic replanning etc. For a complete picture of the system architecture turn to [Brown et al.].
Secondly, the route chunking mechanisms, using the plan, the user model, and the map database, processes the plan into suitable chunks according to the results obtained in our study. It will in the tourist case change the plan into a “Road1 Intersection1 Road2, Road2 Intersection2 Road3” pattern where the intermediate road is repeated. It will then adorn this pattern with names of roads and intersections, landmarks, lane information, etc., where it is needed. Placement and maintain instructions will be added where needed.

In the citizen case, the route chunking mechanism will change the plan by deleting all the intersections and as many roads as possible. It will also serialise the objects to be described into chunks of reasonable size, and decide which roads to substitute with landmarks or nicknames. Here we will use the same principles as did our subjects, see section 4.3, for instance; which roads to delete will be deduced from their hierarchical level and place in the pattern; all the intersections will be deleted; if an extremely low level road has been chosen, we will describe it if comes in the middle of the trip; the first and last parts of the trip will be taken away, etc. We shall look out for any strange twitch in the plan where we might need to use the mixed description rather than the declarative description. The roads are given names taken from the map database.

Lastly, the output from the generator is fed into a grammar. In view of the fact that the descriptions constitute such a limited subset of natural language, as can be seen from our descriptive grammar, even a relatively small and unsophisticated grammar can be expected to produce satisfactory results.

![System architecture for the route chunking mechanism.](image)

The output from the route chunking mechanism, is roughly equivalent to a message as defined by [Mellish and Evans]. It is a nonlinguistic object, at a more abstract level than the linguistic level. A message is built from a limited message language specifically devised to express a certain domain, in Mellish and Evans case, the planning domain.

Our *message language* is much more limited than [Mellish and Evans] since it only will deal with sequential plans instead of nonlinear, there is only one actor able to perform the instructions instead of several, and the utterances we would like to express are fewer. Still we would like to make our message language rich enough to later on be able to express explanations, and as a basic datastructure for some reasoning when we have dynamic replanning\(^8\).

---

\(^8\) Dynamic replanning comes into play when the system can replan due to traffic intensity or other variables, a new plan sometimes needs to be presented to the driver. Some reasoning as to how much of this new plan needs to be presented, has to take place.
A message is firstly an utterance, but also various ways of combining messages. An utterance in turn is either an instruction to perform an action, an explanation of an action, or an expansion of a complex action into subactions, etc. An action in our case is what the driver can do while following the route, i.e. turn, maintain, placement or drive actions. We have gathered a suggestion as to what structures might express this in figure 9.

The purpose of the route chunking mechanisms will therefore be to take a plan and decide what is to be said, in what order and to whom, and then produce a message with that information.

In figure 8 above, we have only allowed the user model to influence the route chunking. User modeling is traditionally still regarded as important only for the user interface. It is important to point out that in this case, and many others, a user model should really be allowed to influence earlier stages as well. In this case, it would mean influencing the planning process. If the planner can be adopted for the tourist, so that it produces plans with fewer turns, as in [Elliott and Lesk], and includes roads at higher hierarchical levels, it would indeed be easier to describe the route to the tourist. The citizen would benefit from the shortest route, even if it is hard to describe and follow. In [Lindervall and Höök] we investigate this issue further.

<table>
<thead>
<tr>
<th>MESSAGE:</th>
<th>utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>--- see below</td>
<td></td>
</tr>
<tr>
<td>time-then(MESSAGE,MESSAGE)</td>
<td></td>
</tr>
<tr>
<td>--- two bits produced in sequence, this indicating time order</td>
<td></td>
</tr>
<tr>
<td>UTTERANCE:</td>
<td>do(ACTION)</td>
</tr>
<tr>
<td>--- instruction to perform an action</td>
<td></td>
</tr>
<tr>
<td>now(STATE)</td>
<td>--- indicating that some state now holds (explanation)</td>
</tr>
<tr>
<td>expansion(ACTION, ACTION)</td>
<td>--- describing the expansion of an action into subactions</td>
</tr>
<tr>
<td>ACTION:</td>
<td>turn(DIRECTION,INTERSECTION,INTERSECTING-ROADS)</td>
</tr>
<tr>
<td>--- turn in some directions in a certain intersection from one road onto another</td>
<td></td>
</tr>
<tr>
<td>maintain(DISTANCE)</td>
<td>--- keep going on a road for some distance measured in meters, blocks, etc.</td>
</tr>
<tr>
<td>placement(LANDMARKS)</td>
<td>--- you will see certain landmarks</td>
</tr>
<tr>
<td>drive(ROAD)</td>
<td>--- citizen instruction that can be expanded into the other actions</td>
</tr>
</tbody>
</table>

**Figure 9.** Some messages, utterances and actions in the message language.

6. Conclusions and future work

We have exposed the underlying principles of citizen and tourist route description done by humans, both in terms of how they are described and, to some extent, why. We have also seen how the actual route choice differs for the two groups.

Furthermore we have indicated how these principles could be implemented and used as an interface to a route guidance system. We believe that the solution we have outlined is good for citizens, but not so good for tourists, but this needs further inquiries. We intend to make an implementation of the ideas, and to use a real world map. This would enable us to test the obtained route descriptions with humans.

In figure 5 above (page 8) we have summarized the tourist route description language. In principle, it says that:

- The basic unit for describing a route is an instruction to perform an action.
- Actions are choices made in an intersection.
- Instructions can also be placement instructions and maintain instructions.

There are some heuristic rules as to when any of the grammar rules comes into play, like for instance, that when describing an intersections, the name of intersection is most important,
then comes type, landmark and size and thereafter intersecting road and number of intersections.

The mixed description (as said above) can be based on the grammar in figure 5 (page 8), by simply removing all the "IntersectionId" and the "RoadId" from the grammar rule "Action", and take away the placement and maintain instructions.

Let us try and summarize the declarative descriptions. Here the grammar rule for the surface language is very simple, see figure 6 (page 8). What is interesting about declarative descriptions is that they are extremely short. The grammar in figure 6 only explains parts of this, that the instruction is not adorned with so much extra information, and is declarative rather than procedural. The rest of the explanation comes from the fact that roads are left out in the middle of the trip, the start of the trip, and the end of the trip. Those gaps can be expressed in some heuristic rules:

- The first 1 - 3 roads before a road on a higher hierarchical level can be taken away from the declarative description.
- If the goal is on a high hierarchical level, the last 1 - 3 roads can be skipped.
- In the middle of the trip, only roads that help excluding other alternative routes are mentioned, plus some roads that help making the route description 'complete' (these road are usually on a high hierarchical level and it would be odd not to mention them).

There are also some principles for when a switch to mixed descriptions is made. It occurs when a 'strange' route has been chosen. Something seems to be a strange route whenever:

- roads on a low hierarchical level are chosen,
- or when the chosen route is much longer than the "as the crow flies" distance.

Finally, the citizen pattern: “drive Road1, drive Road2 ...”, is only changed using a landmark or hierarchical information on a few occasions. The principles seems to be that:

- A road name can be replaced by a landmark that uniquely determines that road when the subject can not remember the name of the road, or when the landmark is such a known item that its name would supersede the road name.
- Whenever the hierarchical level of the road is unusually low, it indicates that a strange road has been chosen, and to emphasise this the hierarchical level of the road is mentioned.

In terms of route choices, the underlying principles seems to be that:

- Tourist routes are chosen by subjects in order to be easy to explain. A route is easy to explain when it is on a high hierarchical level, contains few turns, and goes by important landmarks.
- Routes are constructed by tracing the route backwards from the goal to a point that is easy to connect with the starting point.

In our study, the subjects were asked to give the route descriptions verbally, and it could be argued that we would have found other principles if they had been allowed to draw maps as well. The decision to limit ourselves to verbal descriptions, is due to the fact that quite a large amount of the population is not very good at interpreting and using maps [Thorndyke and Goldin]. The same conclusion was drawn by [Streeter and Vitello]. We also know that the visual channel is highly overloaded already in the driving situation. If the audible channel is used, the visual channel is not cluttered [Davis and Schmandt]. Still, a similar study where subjects are allowed to draw maps as well, would be interesting.

7. Acknowledgement

This work was sponsored by IT4 research programme, within Prometheus, PRO-ART European Cooperation.

We would like to thank Carl Brown, Annika Wærn, Yvonne Wærn, Manny Rayner, Per Lindevall and Björn Gambäck for many valuable comments on this paper and the study in general. We also thank the 10 subjects who took part in the study. Thanks also to Lynn Streeter and Michael Lesk for taking time to discuss their research results in the field.
8. References

[Alm] “Show me the way to go home - or drive me home old chip!”, Håkan Alm, VTI, Sweden, 1989.


Chapter 9

Human Routes, Routes for Humans

Per Lindevall and Kristina Höök
Human Routes, Routes for Humans

Working Paper PS-9103

Per Lindevall & Kristina Höök

August, 1991

Knowledge Based Systems Laboratory
Swedish Institute of Computer Science
Box 1263, S-164 28 KISTA, SWEDEN
Phone: +46/8-752 1300, Fax +46/8-751 7230
perl@sicse.se, kia@sicse.se

Abstract

A solution as to how to make human adapted route planning in Interactive Route Guidance (IRG) systems is outlined. Firstly, different driver needs is motivated and summarized. A solution using a hierarchical map representation is then presented.

1 Introduction

In available Interactive Route Guidance (IRG) systems, the way the route is planned, is usually to find the shortest route between two points. The shortest route will be the shortest in terms of time or distance. We could claim that if we had a perfect IRG system, the driver should simply sit back and follow the directions given by the system, and she will get to the destination on the quickest or shortest route without having to plan or understand the route herself. Unfortunately, the shortest or quickest route might be horrible to drive. It might contain many left-turns, it might go on small, narrow roads, etc. The driver might also be unwilling to simply follow the direction, instead she will abandon a route suggested by the system if it diverts too much from her own opinion of the best route choice.

Therefore, the total sum of costs for a trip used by the planning-part of the IRG system, must reflect the desires of the driver, apart from reflecting the desires of the society. The society might advocate routes that minimize pollution and/or accidents and so on. It should be noted that the introduction of advanced navigation systems along the lines we are discussing can drastically change the behaviour of both the drivers and the society. The effects of such systems has to be considered and estimated along with their design and introduction.

The behaviour of drivers of today performing navigation tasks has been studied by several researchers. Elliott and Lesk [Elliott&Lesk] report that drivers strategies typically include finding main roads and that they stay on these as long as possible, that is, as long as the general direction towards the destination prevail. They also found that humans dislike computer planned minimum-distance routes because they contain far to many turns. Streeter
and Vitello [Streeter & Vitello] report that the two most important factors to minimize on short trips were road construction/bad roads and traffic. For longer trips the important factors were road construction/bad roads, use of main roads, total distance, good scenery, few stop lights and travel time.

From these papers one can conclude that drivers rate factors that promote comfort and ease of driving higher than total distance and travel time when planning a trip. Some of these factors are static and can be imbedded in a hierarchically organized map database and to achieve routes that reflect the above mentioned preferences is a matter of search algorithm design. Dynamic factors are traffic density, mean speed and travel time. To be able to include them in a route evaluation function they have to be expressed explicit for each segment. For the events and conditions that effect dynamic factors, such as weather, an area representation [Gustavsson & Lindevall, Brown & Lindevall] that work together with a hierarchically organized map [Brown et al B] achieve these means.

From the society point of view, traditional means to achieve traffic control is through directing the traffic to certain roads/streets and limiting the access to others. These means are inherited by the map database and are static. With the area representation a new mean for dynamic traffic control is at hands. As the situation is today certain city regions have traffic information broadcasted from a local radio station. To be able to use the information broadcasted, the driver have to stay tuned to a specific radio station, she have to understand what is said and she must have reasonable good local knowledge to be able to reason about the effects of the message. This implies that there is a great chance that local traffic messages are not to any help for visiting drivers. The area representation can express the desires of the society in a transparent way to the driver that is unaffected by her choice of radio station.

In this paper, we shall go through some of the criteria used by humans today when planning routes, some studies by others (chapter 2.1), one study by us (chapter 2.2), and some investigations into the the needs of special groups (chapter 2.3). We shall also look at the few reports that have investigated how humans react to computer-planned routes done on existing IRG systems (chapter 2.4). In chapter 3, we then outline a solution as to how both the society and human criteria of route planning can be taken into account by a planner based on a hierarchically organised map database.

2 Human Route Choice

In this chapter we investigate the literature on what have been said about human route choice. When reading the different reporters the reader shall bear in mind that different cities have different characteristics which can produce seemingly different result such as the investigations by Chase and Pailhous. In an abstract sense Chase and Pailhous came to the same result but Pittsburg do not have the characteristic primary and secondary street network as Paris do. The investigation have been extended with an experiment of our own and as a closing remark we point on necessary properties a route shall have for different user groupes.

2.1 Other studies

There are some studies on how humans choose routes. These investigations into trip planning are mainly concerned with how people plan routes in terms of which criteria they find important and how they search the map, or their memory to construct the plan. Investigations into how to design trip planning systems from a cognitive point of view was first investigated by [Elliott & Lesk].

Elliott and Lesk implemented a system, in which they could, given a starting point and a destination, plan a route between the two points. They have implemented several algorithms in order to test their feasibility; breadth-first/depth-first search, pre-storing important routes, divide-and-conquer, and keeping a hierarchy of maps with progressively fewer streets.
Feasibility was tested both in terms of fastest algorithms, but also in terms of the route being understandable to humans, which is what we are interested in here. People’s strategies seemed to be first finding any main roads, and then apply divide and conquer as well as depth-first search.

The experiment set-up consisted of eight subjects. They were asked to sit down with a map over an unfamiliar area and decide upon routes between two points. In general people first searched for an important road going in the right direction. Then they did depth-first search, single ended, going to and along important roads. A first-hit strategy was used, they did not look for a second route. Interesting enough, people tended to depend upon colour and presentation, not labels, to find main streets.

Elliott and Lesk conclude that a hierarchical search probably will be the best computer algorithm for finding routes that are both short and acceptable to humans. It was shown that the theoretically best algorithms, produced routes that took too many turns, and they were also unfeasible since they were designed to handle the worst-case, rather than the normal case. When they put on an extra cost for left turns, the algorithms tended to render better algorithms.

In another study made by Streeter and Vitello [Streeter & Vitello] they found that a good predictor of whether people would chose a particular road was whether the sum $A + B + C$ (were $A$ equals the straight-line distance from the start to the road, $B$ equals the distance traversed on the big road, and $C$ equals the straight-line distance from the departure point on the road to the destination) did not exceed the straight-line distance between start and destination by more than about 20%.

They also showed that residents used primarily the local road system, whereas experienced subjects (living in the area) used secondary roads and major roads, and tourists used mostly major roads. Humans seemed to follow a hierarchy of the road net, where they tried to find a road at one of the levels in a global way, i.e. looking at all the roads at that level. All the algorithms on the other hand, used a local way of searching a particular area.

Humans tended to switch to a lower level in the hierarchy when the difference between $A + B + C$ (see above) and a straight-line distance exceeded 20%. For tourists, a difference of 30% is acceptable.

In a follow up study made by Streeter [Streeter] alone, she investigates how we can find routes that people that are unfamiliar with an area find usable. She has studies experienced rental automobile employees, who characteristically give customers directions. She compared routes generated by her own heuristics to the automobile rental employees choices.

There are two interesting properties with Streeter heuristics. The first is that it tries to have a birds-eye view when planning the route. She says that it is reasonable that humans do look further than to the next intersection when they plan a route, and that a more global view is necessary. In fact, other algorithms only in 5-15% of the cases succeed in matching humans choices, while her algorithm accounted for 52% of tourist routes, 33% of experienced and 17% of resident routes.

The other characteristics of her algorithm is that it uses a three-level hierarchy of the city. One level is the local road system, the second is the secondary roads and major roads, and the last level are major roads. People seem to try and solve problems at the highest level first, and then going down in the hierarchy (see above). As expertise evolves, this structure is flattened, so that an resident searches through the entire network, instead of just one level.

In this study, Streeter again finds that people tend to stick with as high a level in the hierarchy as possible, even when this leads to a considerable loss in terms of distance.

Both [Chase] and [Pailhoux] have made investigations into how taxi drivers learn and make use of their knowledge of large-scale environments.
Pailhous divided the streets of Paris into a 2-tiered hierarchy: a base and a secondary network. The base network was defined as those streets that were high-lighted on the Paris map, about 10% of the streets. The secondary network became the rest. Pailhous studied experienced and novice taxi drivers and found that both groups used the base network. When presented with a detour problem, Pailhous found that half of the expert taxi drivers would chose a route through the secondary network to get around the barrier in an optimum way. Novice taxi drivers on the other hand, would select a longer base network route to get around the barrier. Pailhous concludes that the basic strategy of taxi drivers is to get to the base network as quickly as possible, and then stay on it for as long a possible.

Chase has studied Pittsburgh taxi drivers, and found results that to some extent contradicts Pailhous. The expert taxi drivers in his study, did use the secondary network whenever possible, instead of sticking to the base network. Chase gives two possible explanations to this: a) there is no hierarchical division of the street system (with the exception of the Parkway in Pittsburgh) and streets vary on a continuum of familiarity, or b) the preferred street system, or base network, expands with expertise.

Chase furthermore argues that taxi drivers do not navigate by means of a map in the head, instead the underlying representation of the city is hierarchically organised so that locations are nested within large regions and neighbourhoods, neighbourhoods are nested within larger regions and larger regions are located with respect to more global features. Chase suggests that the hierarchical storage of places and streets also offers economy of storage, and that it is an integrating part of planning a route. To get from one location in one neighbourhood to another location in a different neighbourhood it is suggested that the driver first finds a route that connects the two neighbourhoods, and then the rest of the route is either subsequently generated or it is filled in as the driver goes along. The driver can continue to follow the “global” plan until cues from the environment are encountered that trigger specific routes at choice points along a route.

Underlying human route choices are human capabilities of representing and processing spatial information. Kuipers [Kuipers78, Kuipers82] is interested in the stages that humans go through when forming a mental map. He says that we should go back to a notion of two different representations. One more fundamental representation that is learnt easily by children, and another that you need to train yourself to be able to do. The first representation simply involves the ordering of objects you pass on, for example, a path, the connectivity and containment. Whenever passing an object this gives the clue to what we expect next and what action to perform right now. So that we remember when seeing the hedge, that it is this corner where we should turn left. We do not have a mental representation of the entire route in our head that we can reconstruct in our mind, but we rather have a list of things that comes alive, part by part when passing objects in the list. In this representation we are unable to measure time and distance or to relate objects not visible for the moment to one another, directions and distances between them etc. This representation might explain why people are sometimes unable to go from B to A even if they know how to get from A to B.

Kuipers claims that if we really have a map representation in our head, we would be able to relate any object in that map to any other object in the map. Since we frequently are unable to relate one region to another, but might have a perfect orientation within the regions, we are sometimes unable to judge correctly how an object in one region relates to another object in another region. Kuipers proposes that modifying the map-in-the-head-metaphor into an Atlas-metaphor where one region is represented as one page in the Atlas, and two regions are related only at a few point where they connect, would be a better metaphor. So instead of assuming the existence of an isomorphic function that converts all points in reality into point in the head, and that possible gets some of these points wrong, we can predist that an entire region might be for instance, a bit rotated compared to another region. This would explain why all the points in one region are distorted in comparison to the other region.
In [Gärling et al.], the development of mental maps is also investigated. They show that, contrary to the prevailing hypothesis, people seem to learn paths between landmarks, before they learn the relative locations of landmarks. More important to us, is that they show that the relative directions of landmarks is quite easily learnt, but that the relative distance to landmarks is much harder to learn. In another study by [Sadalla & Magel] subjects estimated distances along a route containing varying numbers of right-angle turns. Based on memory of the route, estimates of distance increased with the number of turns. Subjects seemed to simply sum up the number of segments travelled between turns, rather than estimating the length of those segments and adding them up. It is also known that peoples estimates of distances from x to y sometimes is different from their estimates of distance from y to x. Similarity they sometimes break the rules which says that the sum of the distance from x to y and y to z is equal to or bigger than the distance from x to z.

2.2 Our study

In a study made at SICS [Höök & Karlgren], we were mainly investigating route descriptions, but we were also able to extract some results as to why sometimes different routes were chosen. The experiment was set up with 10 subjects, all of which were experienced drivers in the city chosen (Stockholm). They were asked to mentally, or by map, find a route between tow spots in the city for 6 pairs of starting and end points. They were then asked to describe the routes first to a tourist and then to an experienced driver in the city (hereafter called the resident). They were encouraged to chose different routes for the tourist and the resident whenever they felt it necessary, and also to tell us why they choose different routes in these cases.

The six pairs of starting and end points were chosen in such a way that, one of the pairs did not allow different route choices, but the other ones did. The pairs were also chosen so that the possible routes between two points would differ in terms of patterns, where a pattern can be thought of as for instance, quickly getting onto a big road, staying there as long as possible, then getting off it onto smaller roads, another pattern being only travelling on small roads.

We found that in quite a few cases, the subjects would chose different routes for the tourist and the resident, see table 1. Usually, when they choose different routes they would mention the fact that they did so, and sometimes why. Usually the reason was that the route chosen for the tourist was more easily described.

Now, what makes a route more easily described than another one? [Davis & Trobaugh] have defined some properties of easily described routes. They say that an easily described route should not contain too many turns, and it should, when possible, go via important landmarks. One of our subjects explicitly verified the second property. He consequentially tried to get the tourist past big landmarks that were visually attractive. The first hypothesis could also be verified. The tourist route was usually more straight ahead, included fewer turns.

<table>
<thead>
<tr>
<th></th>
<th>Road1</th>
<th>Road2</th>
<th>Road3</th>
<th>Road4</th>
<th>Road5</th>
<th>Road6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Different routes</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

1 From one interview: "I am thinking about whether I should take... try to explain the same road a gave Jussi, or if there is any easier one. It is not certain that ... It is not certain that the shortest route is the best to explain, right?"
Table 1. Numbers of times different routes for chosen for the resident and the tourist.

We would like to add another property that has to do with hierarchical level of roads. Our subjects frequently tried to chose roads that were on the highest hierarchical level when possible. The concern seems both to be that the route should be the shortest in terms of time, but also that, especially tourists, should not get lost. A bigger road is easier to recognize, it has better signs, and you can usually travel on it longer than on a road on a lower hierarchical level.¹

In table 2 we have illustrated the difference between the tourist and resident routes going from Katarina church to the university in Stockholm in those cases where different routes where chosen by the subjects. The numbers refer to the hierarchy of road types described in section 2. You can notice that the number of roads appearing in the resident route are slightly higher, but the interesting difference is that the hierarchical level is in general lower for the resident.

<table>
<thead>
<tr>
<th></th>
<th>#obj</th>
<th>descr-obj</th>
<th>Hierarchical level of the roads involved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject#1: Citizen</td>
<td>12</td>
<td>4</td>
<td>4-3-2-3-3-2-3-2-2-1-4</td>
</tr>
<tr>
<td>Tourist</td>
<td>10</td>
<td>10</td>
<td>4-3-2-1-1-1-2-3-1-4</td>
</tr>
<tr>
<td>Subject#2: Citizen</td>
<td>10</td>
<td>5</td>
<td>4-3-1-1-1-2-3-2-1-4</td>
</tr>
<tr>
<td>Tourist</td>
<td>9</td>
<td>8</td>
<td>4-3-2-2-3-2-2-1-4</td>
</tr>
<tr>
<td>Subject#3: Citizen</td>
<td>12</td>
<td>5</td>
<td>4-3-2-2-1-1-2-3-3-2-1-4</td>
</tr>
<tr>
<td>Tourist</td>
<td>10</td>
<td>10</td>
<td>4-3-2-2-1-1-2-4-1-4</td>
</tr>
<tr>
<td>Subject#4: Citizen</td>
<td>16</td>
<td>8</td>
<td>4-3-2-3-3-2-2-2-2-3-2-1-4</td>
</tr>
<tr>
<td>Tourist</td>
<td>10</td>
<td>8</td>
<td>4-3-2-2-1-2-2-3-1-4</td>
</tr>
</tbody>
</table>

#obj = number of roads in the chosen route, descr-obj = number of described roads

Table 2. Differences in number of roads, number of described roads and hierarchical level of roads in tourist and resident (citizen) routes.

Different routes were also quite frequently chosen for goals that were along the same line, but where one goal was much further away than the other. An interesting pattern becomes apparent. When a route is chosen, it seems like humans first try to backtrack from the goal in the direction of the starting point, to a well-known spot or road to which they know that they can find their way from the starting point.

In figure 1, we see that the one of the goals that we chose for our study, lies very close to a big road, marked as Essingeleden. This big road was very frequently chosen for the second goal, Drottningholm, but not as often for the first goal, S.t Görans hospital, that lies very close to Essingeleden. We found that since chosing this road would mean going in the wrong direction a couple of blocks in the direction from the starting point, subjects were unwilling to make this connection, both for tourists and for residents. Instead they connected the point Fridhemssplan with the goal, and tried to find a route from the starting point to Fridhemssplan. The route through Fridhemssplan is much more complicated both to describe and to ride.

¹ On the other hand, one person rejected a bigger road for esthetical reasons. He felt that a tourist should go on beautiful roads, and also that a big road has the disadvantage of being to hard to get off at the right place. If you miss an exit you might sometimes have to travel several kilometres before you can get off and turn back.
Essingeleden is a route at the highest level of hierarchy, and the speed-limit is here 70 km/h instead of 50 km/h which is the normal speed limit in the city.

Figure 1. Going from Karlaplan to S:t Görans hospital or Drottningholm.

Another pattern was found with the second pair of goals that were situated along the same road. The closest goal, Ingmarsgatan, was this time very close to the road that people normally used for the second goal, Stockholms University. Different roads were still chosen for the two goals, but the consideration did not seem to depend upon the fact that you would have to travel one block 'back' from Roslagstull, in order the reach the first goal from the chosen road. Instead the ability to describe the road in an as unambiguous manner as possible was the criteria of whether one route would be chosen before another. One route is much more difficult to describe since it involves more objects, and the objects are at a lower hierarchical level. As a consequence this route was more seldom chosen, and usually only chosen for the resident.

2.3 Groups with special needs for planning

We shall not go into any great detail about the needs of various user groups concerning the way the route should be planned, but only show one example of a group of drivers that certainly will impose very specific criteria on route choices. Another group that has been somewhat considered in other work [Parkes et al] is the elderly drivers. They find certain roads and situations very stressful, and might therefore need another set of principles for route planning, at least for certain parts of the route.

One group that has been investigated by [Gray et al], are the commuters group. In a study by them they have tried to define the functional requirements for an advanced driver information system (ADIS), through an extensive study and interview with commuters. The study is quite interesting since it is one of the few investigating commuters requirements from a computerized traffic information system.

The study is divided into three parts, and conducted in Seattle, Washington, where there are huge congestion problems. The first part of the study was a questionnaire with 3 972 respon-
dents. The purpose of the questionnaire was to determine what kind of groups of commuters could be found. Four groups were distinguished that differ in the amount to which they can accept to change route, or time to travel.

From the 3,972 subjects, 96 were chosen to represent the four groups, and were interviewed. The interview shows that commuters are familiar with alternative routes, but they rarely use them, and whenever they do use them, they experience more stress than when they travel on their primary route. It also seems important to see the actual congestion before changing route, and commuters do not seem to believe in the available traffic information. From this questionnaire the authors also conclude that traffic information needs to be specific in term of time and geography.

The purpose of the study we somewhat different from our purpose, but we still can draw some conclusions concerning the needs of the commuters for route choice. It seems obvious that the normal route to and from work must be stored and known by the system. Any change of route due to the dynamic traffic situation, must then consider some factor as to how much better this new route has to be in order to supersede the normal route. When calculating the route, it will also be extremely important to be able to show the gain in terms of time in order to convince the commuter.

This group of driver, should, of course, be studied more extensively before any definite conclusion can be drawn as to which criteria should govern the planning of the route. It is a good indicator though, of the fact the different groups shall probably have different criteria at in some respects and that those need to be considered when planning.

2.4 Evaluation of existing IRG planning

In a paper by [Bonsall&Joint] some conclusions from one study are on an existing IRG system (LISB) and another study on the so-called IGOR system is reported.

First, a short review of the LISB system [von Tomkewitsch]. User's begin by keying the location of their intended destination into an in-vehicle unit. A small screen then displays the crew fly direction and distance to the destination (known as “autonomous mode”). When the vehicle passes a roadside beacon, the display changes to “full guidance” mode. From this point on, LISB's calculation of the minimum time route to the destination is made known to the driver via symbols on the display and audible messages instructing the driver to make the requisite turning at each intersection. Towards the end of the journey, the system reverts to autonomous mode. (The autonomous mode is due to the lack of enough beacons, or rather possibility to handle more than a limited number of beacons).

In the study on the LISB system, Bonsall and Joint found that the use of the system seriously goes down after a short time of use. The main obstacle for using the system seems to be the way that the destination is given to the system. The study also shows that even in those cases were people actually do use the system about 30% will still not follow the instructions given by the system. The reasons for not following the instructions were according to the subjects that:

- they thought that the system was sending them in the wrong compass direction,
- the advice was given too late,
- the system suggested them to leave a route which was normally good and which had no obvious problems the day in question,
- it suggested they use a route that was normally very congested,
- the system was apparently malfunctioned,
- it sent them in a direction contrary to the roads signs.
The authors conclude that all of these demonstrate that, unless guidance is backed up by other information available to the user, either from her experience or from her direct observation of prevailing circumstances, she may well decide not to follow the advice given.

The authors further conclude that drivers are unlikely to request guidance if they find the effort of doing so out of proportion to their perceptions of the benefits to be gained, and that they are very likely to ignore or reject guidance advice of they do not find it credible. Factors affecting its credibility include:

- the extent to which it is corroborated by, or in conflict with, local evidence about the alternative (Eg the alignment with respect to the destination relative levels of congestion),
- mismatch between the system’s route choice criteria and those of the driver,
- the driver’s familiarity with the local network,
- the quality of advice previously, and particularly very recently, received from the system by the driver (objective measures of previous quality seem a reasonable proxy for what is no doubt a much more complex phenomenon including factors such as the frequency of obvious malfunctions, etc),
- the driver predisposition to accept/reject advice.

The last point above comes from the second study, where the interviewers were able to classify the subjects into four groups:

(i) dissenters (about 10% of participants) who object to guidance on ground that it is an intrusion on personal freedom;
(ii) the undecided (about 40% of participants) who are torn between a desire to be guided and a cynical mistrust of the system;
(iii) the prudent conformists (about 40% of the participants) who will follow advice if, and only if, it is logical;
(iv) the trusting (about 10% of participants) who are unsure of their own skills and, lacking experience, are happy to put themselves in the system’s hands.

2.5 Summary of human preferences

Even if humans chose routes in certain manner, we could decide that the IRG system still should try to find the best route, in terms of time or distance rather than adapting to the human preferences. The argument against that is threefold:

- first, some groups of driver will refuse to or dislike driving the routes suggested by the system. The routes chosen might be the best in terms of time or distance, but they will not be perceived that way. Rather the drivers will be suspicious about the route choice and end up driving their own choices of routes.
- second, a route choice that only is optimal on time or distance will sometimes be hopeless to drive. It might contain many left turns, or put a tourist in a situation where a misinterpretation of an instruction will put her in an awkward position.
- third, the route choices has to be communicated to the driver and if they adhere to her mental image of the road network of a city and how to traverse it, it will be much easier to explain them and for the driver to understand the description and form a mental model of it.

Therefore, we might decide that adapting to human preferences is a good idea. So what human preferences can we extract from the studies and theories described above? Firstly, it seems to be the case that drivers with different purposes for using the IRG system, and drivers with different amounts of knowledge of the city will have different preferences of route choice.
A general conclusion could be that the less knowledge of the city, and when the purpose of using the system is only to reach the destination at all, the more the IRG system might be allowed to, or forced to, chose routes that are suboptimal in terms of time or distance:

- In Streeters A + B + C algorithm the IRG system should try to make the A and C distances short.
- Turns should be avoided, especially left turns.
- The hierarchical level of roads involved should be as high as possible, or in other terms: we should stick to the basic network whenever possible.
- Sometimes roads should be chosen only because they go past important and interesting landmarks or other recognizable items.

For the professional driver, it is important that the IRG system is able to give a better route choice than the driver would chose herself, the problem will sometimes be to convince the driver that this actually is done. Since humans are so bad at estimating distances, and understanding the relationships between two well-known areas in the city, this can be quite difficult. For instance, when the route chosen will tend to take the driver to a spot behind the goal, a spot not so easily connected to the starting point, we shall need to convince the driver that this is actually a better choice. The professional drivers will on the other hand allow the B of the A + B + C algorithm to be smaller or even obsolete to a bigger extent than the tourists. Therefore we shall not have to consider whether there are too many turns in the route. It does not either have to be easily described. For a descriptions of routes aimed at professional turn to [Höök & Karlsgren].

For the commuter, the situation again is different. She will experience more stress if she has to leave her usual route, mostly due to not knowing for sure that there is a profit from chosing another route. It might therefore be necessary to have a threshold for when another route is chosen, so that it only happens when there is a substantial gain in terms of time or distance. To do this we need to store or somehow remember which is the normal route for the commuter to and from work. As said above, the IRG system must also have some means of convincing the commuter that the IRG route choice actually is better.

We can draw the conclusion that it is necessary for the map database to be organised in such a manner that we can mimic the human learning of spatial information. This would mean constructing a hierarchical structure on which algorithms like Streeters A+B+C algorithm can be used.

Some of the points from Bonsall and Joints paper might be corrected by giving more information than is available from the LISB system. For instance, the reasons for choosing an unexpected route might be given. This would to some extent remove distrust in the system like when suggesting a route that is usually very congested, or when leaving a normally good route, etc. Issues of presentation is tackled in [Höök & Karlsgren]. The other points made in Bonsall and Joints paper may partly be tackled by adapting the actual route planning to human preferences.

3 Planning routes that people like?

In this chapter we outline an attempt to realise the conclusions from section 2.5 in an implementation using a hierarchical map structure as described by [Brown et al B] and adapted search strategies better suited for humans. The solutions described below build upon an already existing implementation, where the hierarchical map datastructure is used together with a normal A* search algorithm.

3.1 A Hierarchical Map Structure
The hierarchical map representation format proposed in [Brown et al B] is designed to mimic the map-makers use of road classification and map scale. The map scale is used to suppress information to a certain degree when a trip is first outlined, that is, roughly connecting an origin with a destination. At a later stage when more and better detail is needed about a specific part of the route the information is found in more detailed maps. Map-makers classify roads by printing "better" roads (as defined by a road administration) in a more eye catching manner. Thus it is quite easy to find a route if one first look in a large scale map for the general routes and direction and then look for detail in small scale maps once the prevailing direction is decided.

The hierarchical map format consist of three primitives that can be used recursively. The primitives are segments, that connect nodes, and two types of nodes, intersection nodes and expandable nodes. Intersection nodes have a geographical position, a route-list (possible ways to traverse the node) and a identifier. Intersection nodes represent real intersections. Expandable nodes on the other hand represent entire regions of segments and nodes, and that is segments and nodes that are conceptually on a lower hierarchical level than the expandable node. The expandable node have a identifier, a route-list (showing possible connections through the region, by means of exit and entry node pairs) and a position (a "centre of gravity"). Finally segments represent streets or roads between intersections and/or regions, thus a segment has an entry node and an exit node (of either type), an identifier and static properties such as length, speed-limit, classification, etc. With these three primitives a map hierarchy can be built using road classification and map scale as exemplified in figure 2, below.

At the top level ("sverige") only freeways and motorways are found, that is, segments and intersection nodes representing them. At this level all other roads and streets (nodes and segments) are grouped into suitable regions and assigned to expandable nodes representing these regions. For each expandable node at this level the procedure is repeated for its topmost roads and streets. Leftovers are again clustered together in expandable nodes and the procedure applied over and over until no more roads and streets are left to represent. In this way all streets and roads are actually implicitly represented at the top hierarchical level. In the following hierarchical levels, all streets and roads but freeways and motorways are represented implicit or explicit.

With the representation described above every segment (part of street/road) or node (intersection or expandable) will have a hierarchical address that verbally define the geographical origin of the segment or node completely. As an example from figure 2, Ola Hansson gatan in Kristineberg in Stockholm will have the complete hierarchical address;

\[
\text{sverige:svealand:stockholm:kungsholmen:kristineberg:ola_hansson-gatan} \quad (N1)
\]

and Bellmansgatan in Göteborg will have the address;

\[
\text{sverige:götaland:göteborg:vasa:vasastan:bellmansgatan} \quad (N2)
\]

From these two examples one can see on the addressees that Ola Hansson gatan and Bellmansgatan are both in Sverige, but Ola Hansson gatan is in Svealand and Bellmansgatan is in Götaland and therefore cannot be in the same city. Further more one can conclude that a path connecting the two streets must be on the topmost level in sverige as götaland and svealand are two expandable nodes on that level.
Figure 2: The Hierarchies of a Map

Any connection between two nodes on the same level must be found on that level or a higher level. To illustrate the connection between hierarchical addresses and physical connections between areas suppose that we want to go from Flemminggatan on Kungsholmen in Stockholm to Ola Hansson gatan also on Kungsholmen, i.e. example (N1) above. The origin will have a complete hierarchical address as;

`sverige:svealand:stockholm:kungsholmen:kungsholmen:flemminggatan`  \(\text{ (N3)}\)

Comparing (1) and (3) one find that the first four hierarchical levels are identical;

`sverige:svealand:stockholm:kungsholmen:`  \(\text{ (N4)}\)

this is also the address to an expandable node with the same name. The fifth hierarchical level is unique for both example (N1) and (N3) as shown in example (N5) and (N6) below. Thus (N5) and (N6) are expandable nodes under (N4) and any path connecting the nodes (N5) and (N6) must be found on the level (and in the area) defined by (N4).

`sverige:svealand:stockholm:kungsholmen:kristineberg:`  \(\text{ (N5)}\)
`sverige:svealand:stockholm:kungsholmen:kungsholmen:`  \(\text{ (N6)}\)
Mind you that example (4) is one expandable node of many that make up Stockholm and it can be viewed as an area or region in the city of Stockholm. Examples (N1) and (N3) are again nodes under (N5) and (N6) and a path from (N3) to (N1) must go from (N3) through one of several exits in (N6) to (N4) and further down one of several entries in (N5) down to (N1). How this is done will be described next.

The hierarchical map definition conform to the view Chase, Pailhous, Elliott and Lesk have on the "basic road network", "primary and secondary network" etc., and is the necessary basis for implementing the search strategies described below.

3.2 A Search Strategy

The basic search strategy adopted for our hierarchical map is Streeter's A+B+C algorithm combined with Elliot and Lesk's divide and conquer algorithm. What to put instead of A, B and C will depend on what user group the driver belongs to.

According to Streeter and Vitello A and C are: the straight line distance between origin and base network, and the straight line distance between the base network and the destination respectively. B is the actual travelled distance in the base network. The sum A+B+C shall not be more than roughly 20-30% longer than the straight line distance from origin to destination. If it is, B in the base network would be "shortened" or abandoned altogether.

Elliott and Lesk found that an algorithm that first found a main road and then used divide and conquer together with depth first search on minor streets is close to how humans find routes. According to the authors, divide and conquer require that a midpoint can be found efficiently, which is hard in a ordinary street grid. The depth-first search algorithm was set up to follow a street as long as the distance from the current position on the street to the destination (or intermediate destination) decreased. Thus in this way the algorithm avoid turning in intersections. The authors also investigated a breadth-first algorithm that used a penalty in each node based on distance to target in order to guide the search in the right direction. The breadth-first algorithm was generally less efficient than the depth-first algorithm but the different heuristics used provide different heuristics not only inherited from the basic differences in the algorithms.

User groups that we have found so far are Tourists, Residents, Professionals and Commuters. There might be more groups, or rather subgroups of these, like elderly drivers, which are not investigated here. Below, we have put Tourists and Residents (index TR) in one groups since they have similar preferences and Professionals (index P) in another. A third group are the commuters with a little bit different needs from the other two gropes. Firstly, a commuter is either a Tourist/Resident or a Professional. Secondly, the Commuters preferences is different from any other group in terms of when and why to leave her ordinary route, i.e. when to re-plan. Commuters will be given different and enhanced information rather than a special route planning algorithm.

The basic difference between the first two gropes in terms of A+B+C is:

- $B_{TR} > B_P$ and $B_P \geq 0$, i.e. The Tourist/Resident spend generally longer time on main roads while Professionals may avoid main roads completely.
- $A+B_{TR}+C \leq 1.2*OD$ for Tourists and Residents. Here OD is the straight line distance between origin and destination. A is the straight line distance from origin to main road/street and C is the same measure from the main road/street to the destination.
- Tourists and Residents avoid routes with left turns
- $A+B_P+C$ = minimum time (and to some extent maximum distance) for Professionals
- Professionals put an emphasis on A and C (the secondary network) as B could be omitted altogether.

The example from section 3.1 could be interpreted for the Tourist/Resident group as follow:
• A is the distance from Flemminggatan to the nearest exit in the expandable node (N6),
• C is the distance from the entry to the expandable node (N5) that is closest to Ola Hansson gatan,
• B is a path from the exit in (N6) to the entry in (N5).

If the sum A+B+C is larger than 1.2*OD then change the exit and entry points until the sum qualify for a path candidate. When qualified, paths that connect from the origin to B and from B to the destination must be found. Thus the final path is:
\[ A'+B+C' \geq A+B+C \]

if \( A' \) is the real path and not the straight line distance as A is. The same applies to C and C'.

The same example applied to the Professional user reads as:

• A is the distance from the Origin (N3) to the exit in the expandable node (N6) that is closest to the destination,
• C is the distance from the entry in the expandable node (N5) (closest to the origin) to the destination (N1),
• B is again the shortest distance between exit node and entry node in level (N4).

In some cases, although not in this one, the exit node and the entry node in B could be the same node. The path B will in that case contain only one node with length \( \approx 0 \) and thus the distance in B is 0. With the superscript having the same interpretation as in the previous paragraph and the subscripts from above the following relations will hold for the same Origin-Destination pair:

- \[ A'_TR + B_{TR} + C_{TR} \geq A'_P + B_P + C'_P \]
- \[ A'_TR < A'_P \]
- \[ B_{TR} > B_P \]
- \[ C'_TR < C'_P \]

The differences are mainly due to how the intermediate nodes (the entry and exit nodes) are chosen for B.

From our own study [Höök&Karlgren] we saw that people tend to pick a known point close to the destination and plan "backwards" from there to the origin. This known point is always between origin and destination, and never past the destination. This lead to routes that was much harder to drive and more difficult to explain than routes that took the driver on a major road past the destination and then back on minor roads. The strategy for \( B_{TR} \) can very well "go past" the destination in its effort to minimize A and C. If this behaviour is not desired as it may not be for the Resident group, the two sets of exit and entry nodes could be restricted to only contain nodes that are between the origin and the destination. For \( B_P \) "go past" is not the problem but rather choosing exit and entry node that minimize \( B_P \). Again the two restricted sets of exit and entry nodes can be used to find the pair of exit and entry nodes that have a minimum straight line distance. This pair is used to find \( B_P \).

A word about entry and exit nodes. From the definition of a node in [Brown et al B], both node types have a route-list. This route-list define possible ways to traverse the node. Depending on from where you enter the node you will have a choice of possible exits. In intersection nodes the route-list is used to define possible turns and connections through the node. Thus, going north on Odengatan, apart from being allowed to go straight through the intersection one can turn right (eastwards) on Sveavägen but not left. The route-list specifically list what options are possible and allowed in each intersection (a forbidden turn is omitted in the route-list). For expandable nodes the route-list have a little different role. By definition two expandable nodes cannot be neighbours, i.e. their must be at least one intersection node between the expandable nodes. Thus every expandable node are completely surrounded by intersection nodes. These intersection nodes take on the role of a border to the
area defined by the expandable node and together with the route-list (in the expandable node) they express possible entries and exits to and from the area defined by the expandable node.

3.3 Search Algorithms

How the search will work for the various A's, B's and C's is also of great importance in order to receive good acceptance from the different user groups. In general we are looking for the shortest possible route, but with added constraints this route will not be optimum in any way. The constraints added reflect the needs of a particular user group. We now aim to continue and experiment to find the correct balance between how short versus how adapted to the driver the route should be.

For the Tourist/Resident group comfort is more important than than both travel time and distance. Thus, routes have features like (i) stay on the present street, (ii) minimize the number of intersections, (iii) avoid left turns, (iv) easily described. Constraint (i) and (ii) are less important for the B part of the A+B+C strategy. The "easily described" constraint and the "go past" strategy applies only to the tourist group of drivers. (ii) is achieved through the choice of algorithm, while the other features will be decided by the heuristic.

For the Professional driver group, travel time or distance is far more important, thus their routes will always be either fastest or shortest to the target. Implicit in these routes are constraints (ii) and (iii) above as travelling through intersections or making left turns are costly operations and would be avoided to some extent at planning time.

4 Conclusions and future work

We have showed that there is a need to make route planning adapted to human preference. We have outlined some principles that seem to be important. These principles include making different route plans for different user groups as the tourist, commuter, resident and professional driver groups.

Our present goal is to extend the already existing algorithm working on a hierarchically structured database, to take the above mentioned criteria into account. We shall then perform a comparative study on route choices. The comparison will be between humans choices (expert and novices) and algorithmic choices.

5 References


Chapter 10
Dialogue Management and Integrational Priorities in Prometheus Navigation

Carl Brown
Dialogue Management and Integrational Priorities in Prometheus Navigation

Carl Brown

Knowledge Based Systems Laboratory
Swedish Institute of Computer Science
Box 1263, S-16428 Kista, Sweden +46-8-752 15 00

1. Introduction

In our work on the dynamic re-planning of routes we have become aware of the need to integrate the dynamic-replanning functionality with the rest of the navigation system in a way that responds naturally to the drivers requirements for navigational help. In addition, we have become aware of the absolute necessity of integrating the navigation system with all other systems in the car. In this paper we discuss the motivations and the urgency.

In an earlier report [BROW90] we proposed a level of Human Machine Interface Management (HMI-Mgt) as a framework around which Prometheus functions could be integrated. The framework was not worked out in detail for all aspects of HMI-Mgt, but the problem of Dialogue Management was examined in some detail. We briefly review those proposals here.

We shall also propose, at this point, that the concept of HMI-integration be extended to as many of the car systems as possible. Not just navigation functions. Not just Prometheus functions, but all car systems.

2. The Motivation for HMI integration

We can broadly cite two motivations. The first is safety. The second is engineering.

Consider the safety aspect. We are all aware that car systems are becoming increasingly complex and increasingly numerous. With respect to complexity, some Prometheus functions represent systems of such complexity that they will supercede the ability of the driver to make decisions in a complex domain. With respect to the increase in the number of in-car systems, we only have to look in the modern "cockpit" to confirm our suspicion. Currently we can have a primitive navigation system, a multimedia stereo sound system including cassette, CD, DAT and choice of broadcast receivers on AM or FM bands, a telephone, in taxis we also have a dispatch system, various measurement systems relating to the soundness of the car's engines, whether there is fuel, and if there is, how far we can drive on it. What is more, all of this information is delivered through the visual and audible channels primarily with the aid of meters (digital and analog), flashing or continuous lights, lighted symbols, non-lighted symbols, knobs and levers, but also (increasingly) with sound, CRT display, and soon HU display. The important point is that all this information is competing for the driver's attention. In fact, it competes for the driver's attention with the situation outside the car and with the driving task.

The second motivation is an engineering and cost-effectiveness consideration. When complex systems are designed, a system view of the functionality is taken. Integration of parts of the design with other parts is sought. We consider modular design and extendability of systems to be essential. This is especially true in the fields of systems development and software engineering which are in fact the fields involved in the development of dynamic replanning IRG systems and of most Prometheus functions.
Benefits of modularity and integration of modules include: maintainability (subsystems can be maintained without worrying about the effects on the rest of the system), ease of documentation, ease of enhancement (subsystems can be enhanced individually, without effecting the rest of the system and without the need to alter the rest of the system), ease of development (which includes ease of debugging, understandability and efficiency) and verifiability (the communication protocols between modules are verifiable and the operational semantics of module is verifiable). These benefits are gained in the development phase of a complex system and are quantifiable as both development-time and development-cost benefits.

An additional benefit is gained in the production (and hence in the cost) of a complex system. If we consider the in-car electronics and control systems it is easy to see that a system with fewer modules (physical modules and/or software modules) is less expensive to produce. A specific example is the media which communicate with the driver. The driver can only attend to one display area at once. Therefore, there is no need (today) to have separate analog instruments, CRT and other displays. All information presented visually to the driver can be displayed on one device, let us say a high resolution flat screen1 (upon which analog dials, maps, text messages, symbols and even video images can all be presented). In doing so, a great deal of money can be saved on hardware (gauges, lighted symbols viz idiot lights, wiring and control panels). A similar argument pertains with the other channels of communication to the driver, e.g. sound. The driver can attend to only a few sounds at a time.

To reduce the instrumentation and control messages to a single screen and a single sound source requires control of the information stream to the driver. This is an essential part of what we have called dialogue management.

The point must be made that the place to start is with the principles and methods of integration of the HMI function of the multitude of systems. Clearly the systems in a car today are, if taken as a whole, exactly the kind of complex system mentioned above. One can see the start of a programme of system integration in the car industry, but little sign of that integration at the HMI level. Many of the points made above simply repeat the accumulated and accepted wisdom today. It is important to realise that these points also support very strongly the argument that the integration of the HMI functionalities is the crucial problem.

In summary, both safety and cost-effectiveness are supported by full development of HMI-integration. Specifically with respect to HMI, the following points are argued for in this paper:

- Efficient system development is aided with standard interface formats.
- A consistent treatment of time and temporal reasoning is possible with standardised message parameters.
- Alternative sensory media may make use of the same internal message format. Possible benefits exist here in the development of fault-tolerant message systems which might, in the event of faults in one sensory I/O, make use of alternative senses.
- The development of a Driver Workload Model based on the uniform message format and other factors is thought to be possible.

In the next section we discuss the HMI-management and HMI-integration. We pay particular attention to the dialogue management problem and its role in interacting with the Dynamic RePlanning Navigation subsystem.

---

1 One can imagine two high resolution graphics (CRT-like) displays in an automobile. One in the traditional dashboard-instrument panel position visible to the driver and the other visible and accessible to both driver and front seat passenger. Both driver and passenger should have access to some controls/displays such as radio and climate controls.
3. Basic considerations in HMI-Management

A dialogue *message* is the term we shall use for any information passing between the driver and vehicle systems in either direction, in any format. Thus, the messages embody the commands, queries and information that are interchanged. (In several tasks, the order of messages to the driver will be important, as will the location en-route where messages are delivered.) We repeat here the previously stated concern about the ability of the driver to accept messages under certain circumstances. This is the driver-workload factor.

Messages, then:

- move between a diversity of systems and the driver.
- originate from a diversity of controls, systems and interface devices.
- may be presented in a variety of ways to the driver.
- contribute to the driver-workload.
- must be ordered in time and in space.
- have priorities.

The tasks of dialogue management will be influenced by these considerations. They have contributed to the conceptual architecture for the dialogue management system presented below.

3.1 The HMI-Mgt function

Our view is that the exchange between driver and the in-car systems should be controlled by the HMI-Mgt function, providing a single channel through which all messages to and from the driver move. (Figure 1.) By providing a single channel, two design goals are achieved: Consideration of the mental workload of the driver can be made, and ordering of messages can be adjusted to suit the circumstances and priorities. We return to these two points later in this report.

![Figure 1](image)

**Figure 1**
The role of HMI-Mgt

3.2 A view of HMI-Mgt

The conceptual view of the HMI-Mgt process can now be given in more detail. The schema of figure 2. does this. Sub-systems are for the purposes of illustration only and thus remain in less detailed form. The emphasis remains on integration of the view of the HMI-Mgt with the message flow from other sub-systems.

In figure 2, only the minimum set of functionalities is illustrated in the HMI-Mgt function. Omitted, for example, are the Driver Model [SMIL89], a model of the immediate environment [SAND90], and an explanation functionality.
3.3 Integration and the Dynamic RePlanning Navigation subsystem

As seen above, the HMI-Management system exchanges messages with other systems, including among them the Dynamic RePlanning Navigation system. This achieves a division of competence between presentation management (in HMI-Mgt) and presentation planning (in the Dynamic RePlanning system, for example). We use our Dynamic RePlanning Navigation system to see how the presentation planning, which we call HMI-planning, is integrated into that system.

We view the Dynamic RePlanning Navigation system as an integration of four major functionalities. We specific functionalities which we call Dynamic Monitoring, Route-Planning, HMI-Planning and Route Monitoring. These interact with the Meta-Planner to provide an integrated Dynamic RePlanning Navigation System. We are currently examining the interaction of these "modules", the information they exchange and what reasoning about the current scenario (eg. before-the-trip, getting-to-a-main-road, reacting-to-dynamic information, presence-of-a-violation-of-a-time-constraint) takes place in the different modules.

HMI-Planning is a distinct function within the dynamic replanning navigation system. The functionality is to plan the sequence of messages to the HMI-Management system, and to accept messages from it. Thus, another functionality is assumed; that of maintaining a dialogue with the driver through the HMI-mgt, about the route planning task.
3.4 Management of the information flow

The vehicle systems, Prometheus or not, are complex. *This should not prevent the HMI-Mgt process from giving appropriate information to the driver when it is needed.* This is a fundamental tenet of any HMI-management system. Dialogue Management serves this purpose. The dialogue management function must have the competence to manage the information flow to the driver’s advantage. We define dialogue management as the responsibility for directing messages to the correct system, and of supervising the flow of messages; re-ordering or delaying messages with respect to priorities, driver-workload, and time.

The possibility of competing messages makes it absolutely necessary to have a system of dialogue management in the vehicle. The in-vehicle systems interchange messages in complex ways. Because there can be no way of arbitrarily deciding when a particular subsystem will issue a message, management of the presentation of messages to the driver must be provided.

3.5 Division of Competence in HMI-Mgt

The view taken of the division of competence within the HMI-Mgt function is partly evident in figure 4. The Message Presentation function has the responsibility for interpretation of message content with respect to the presentation medium or media. The Dialogue Management Process has the responsibility of ensuring that currently valid messages are available to Message Presentation.

Timing of messages is not a responsibility of Message Presentation. The competence lies within Dialogue Management and the subsystems originating messages. This point is discussed in section 4.1. We have given the competence in *delivery and ordering* of messages to HMI-Mgt, and in particular to the Dialogue Management function.

The competence to *set values for* the parameters (e.g. timestamp, duration, expiry, etc) is given to the subsystem that originates the message. This design decision was made in order to ensure that sub-systems, such as Dynamic RePlanning Navigation systems, can incorporate reasoning about location. Once location criteria are established for the delivery of messages, timing information can be inferred and parameters set.

4. Message parameterisation and message content

The parameterisation of messages is a reflection of the division of competence of the functionalities of the systems. The representation of message content, one of the parameters,
important for specification of the functionalities of the subsystems. The discussion that follows deals exclusively with HMI messages. Similar parameterisation is necessary for messages between other communicating systems.

4.1 Parameterisation of messages (HMI messages)

Messages are passed between subsystems. These subsystems should be viewed as concurrent communicating processes representing co-operating agents in the conceptual architecture. Messages have basic parameters. Individual messages are distinguished by the values of the parameters. Some parameters have already been suggested. What follows is the list of parameters found necessary to date.

**Destination-process**: The process to which the message is directed. The driver is included in the set of possible destinations by the inclusion of a Presentation Management process in the HMI-Mgt model given below.

**Source-process**: The process from which the message comes.

**Type**: Type corresponds roughly to dominating and supporting communication acts\(^1\) in dialogue. Some example values of type are: Request, Confirm, Command, Warn, and Inform. Used by the models of dialogue.

**Priority**: Priority is an indication of the relative importance of a message. Values of priority are on a scale, ranging from least-important to an indication-of-emergency.

**Timestamp**: The point in time when the message originated.

**Workload_Factor**: The contribution of the message to workload (relative to other messages from the Source-process.) The value of the workload the message imposes.

**Workload_Weighting**: A weight given to all messages from the Source_process. This weight is used in the determination of the messages contribution to total driver workload.

**Duration**: The minimum time over which the message should be delivered. Note that a driver has a minimum time requirement [KLAEB90] to become aware of the presence of a message and to interpret it. If such a duration is not available, the message will not be conveyed successfully, even if presented. This parameter is of use when messages are intended for presentation to the driver. It may be useful in other areas (however, see the expiry parameter below).

**Persistence**: A binary value indicating whether or not the message has the property of persistence. A message with persistence is, for example, coolant temperature.

**Expiry**: The time before which a message must be delivered. An indication of the lifetime of a message, the time over which a message is valid. For example, a message about an intersection must be delivered before the intersection is passed. Notice that this requirement is expressed in terms of time at the dialogue management level.

**Medium**: A partially ordered list of the media over which a message may be delivered; from most preferred to least preferred. This parameter interacts with Content in that a composed message may involve more than one medium.

**Contents**: A representation of the contents of a message in a form that can be interpreted by other processes. A semantic representation.

**Opacity**: A message may be transparent or opaque.

The reader will have noticed that one effect of this set of parameters is to give the competence of deciding timing to the system originating the message. Nevertheless, the HMI-Mgt function may choose to delay a message or even not to present it at all. If this is done, it must be done with respect to message priorities and as the result of reasoning about those priorities, timing constraints, and presentation media. Additionally, protocols about undeliverable messages must be developed.

---

\(^1\) Communication acts originated in the concept of speech acts originated by Austin (1962) and extended by [SEAR69]. Recent work by Oim (Universiteit i Tarta, Estonia) in communication acts is perhaps at the forefront of work in this area.
Some information must be pipelined to the presentation. The classic example of this is the speed (usually displayed on an analog speedometer). Pipelined information is treated as in the same way as messages. In this case the pipeline itself has the parameters specified above. (Note that this is a standard computer science technique for communication protocols.)

4.2 Message content

Message content is to be represented as suggested above in a semantic representation common to all systems. This has some obvious advantages. It allows diverse systems to interpret a message, it allows messages to be interpreted in different ways by different systems, and it allows message contents to be compositional.

Compositionality is the ability to represent the composition of sub-messages into larger messages. An example illustrates this. The message content sequence(turn(direction(left), move_to(lane,right))) is a composition of two messages intended to be presented in sequence. Such compositionality has been found to be important in the presentation of explanatory diagrams (see, for example, [STRO89]).

Generality is achieved if all sub-systems can make use of the semantic representation when communicating. The last example might be interpreted as the utterance [Turn left, then move to the right lane] by a Natural Language Generation subsystem of the Presentation process or interpreted as a diagram (graphics display) indicating the instruction by a Diagram Generation subsystem.

Messages can be interpreted in different ways by different subsystems. For example, in the HMI-Mgt schema it is left to the Presentation process to interpret the semantic content and format the message for the specified medium. (Recall that the specification of Medium was a partially ordered list.) This allows the use of an alternative-medium, the next in the list, for presentation when the preferred-medium is currently used by a message of equal or higher priority.

5. The Dialogue Management Process

Message parameters and the HMI-Mgt function having been discussed above, it is now time to turn to the specific question of how a Dialogue Management (see figure 3) process can accomplish the goals of dealing with priorities and timing of messages. This lies at the heart of the HMI-Mgt system.

5.1 An overview of the Dialogue Mgt process

Messages are put into sets in the Dialogue Management process. These sets are ordered and manipulated by the process.

The Dialogue Management process must manage, at any given time, a set of messages to be transmitted: The Pending_Msg_Set. In addition, the process must be aware of the current set of messages that have been transmitted and whose lifetimes have not expired: The Current_Msg_Set. For the purposes of explanation¹, a history of messages transmitted is kept: The History_Set. Other factors are that new messages may arrive to swell the Pending_Msg_Set, messages may be transmitted which reduce the Pending_Message_Set and swell both the Current_Msg_Set and the History_Set. Finally, messages may expire due

¹ Explanation is a desirable feature of reasoning system that communicates with humans. We recognise this, and propose the inclusion of the History Set so that work can proceed in the future on this function. No work is currently underway at SICS on the role of explanation in Prometheus.
to location in time and space which reduces the Current_Msg_Set (and potentially the Pending_Msg_Set).

The Current_Msg_Set is actually a set of partially ordered queues of messages. One queue for each message presentation channel. The queues are queues of unexpired messages delivered to the message presentation process. These messages are presented as soon as they are put in the Current_Msg_Set.

![Diagram](image)

Figure 4
The message sets and Dialogue Management

5.2 Transparent and opaque messages

The Current_Msg_Set contains two conceptually different types of messages; transparent messages and opaque messages.

Transparent messages are those which can be displayed continuously and to which a driver pays occasional attention. Typically these messages contain non-critical information which do not require immediate attention by the driver. Some examples of transparent messages (in vehicles currently on the market) are engine temperature, speed, fuel level, oil pressure, tuning frequency of car radio.

There are times when messages require attention from the driver. In such cases, the messages should be delivered as opaque messages. For example, a driver must be informed that a turn is required at a certain location. In general, navigation related messages will all be opaque. A fuel level message however should only be opaque when the fuel reaches a low level.

The delivery of opaque messages depends upon the media available, and upon the priority of the message. Some guiding principles are:

- Voice delivery is more opaque than dashboard display.
- Heads up display is more opaque than dashboard display.
- A flashing display is more opaque than a steady display.

5.3 The Interpretive Cycle

---

1 The presentation may be interrupted when the interpretive cycle discovers a higher priority message for that channel. The handling of interrupted messages is a subject of investigation.
An interpretive cycle is a convenient way to handle the management of the message sets. The parameters of messages must be used in determining the sequencing in the partially ordered Current_Msg_Set queues. This allows the embedding\(^1\) of a reasoning system within the dialogue management. In this case the reasoning is about time, and about the dialogue. Reasoning is provided by functional models called during the interpretive cycle.

Done this way, dialogue management becomes a question of planning. One possible interpretive cycle is described intuitively below. It assumes a particular division of competence between dialogue management and sub-systems:

1. **Examine Message Sets**
   Remove all expired messages from the PENDING_MSG_SET and inform the originating process of expired/failed messages. Similarly, remove all successfully delivered messages from CURRENT_MSG_SET and inform originating process.

2. **Reason**
   Collect messages from PENDING_MSG_SET and CURRENT_MSG_SET. Order the messages for each presentation medium with respect to time, duration, and priority. Possibly specify alternative media. Propose a CURRENT_MSG_SET\(^t\).

3. **Consult Driver Workload Model**
   Test the proposed CURRENT_MSG_SET\(^t\) on the driver workload model. Accept or reject proposal.

4. **Update Message Sets**
   Update CURRENT_MSG_SET and PENDING_MSG_SET with proposed CURRENT_MSG_SET\(^t\) and remainder of messages.

Clearly there are implementation requirements. The message sets must be buffered by the dialogue management process. A flag must be set when the set is being updated by an external process. Finally, the reasoning system will have time constraints in order to provide true reactive behaviour.

6. **Integration and HMI considerations in Navigation and Planning**

The SICS group and other researchers in Prometheus and DRIVE are examining some of the HMI factors we know to have an effect on the drivers understanding of the navigation and planning system. We can list some of these again. They are:

1. **Driver workload is a factor in the attention a driver pays to messages from any system.** If the driver workload is too high, message content may be lost or ignored. As driver workload increases, increasingly dramatic measures are need to get his or her attention. (Very loud or annoying sounds must be used, for example.) This of course applies to all systems, not just navigation. [REFERENCE ?? PROMETHEUS 2]

2. **Drivers need time to attend to messages, to understand them, to make a decision as to whether action is required, and then to actually carry out the action.** A time of fifteen to twenty seconds has been proposed by [KLÆB90] in research directed particularly towards driving tasks such as turning corners.

3. **The audible channel is most open. That is, we have not yet loaded the sound input to the driver very much.** This means that navigation information can be presented to the driver audibly. We do not yet know whether this is acceptable to all drivers, or even to most drivers. This has been confirmed by recent work at HUSAT [FAIR90]. Note however, that the length of a sentence or utterance directed at the driver has an impact on the time needed to understand it. [VERN90a].

---

\(^1\) See [GEOR89] for a recent discussion of embedded reasoning in changing environments. The question of situated planning arises in Prometheus in this context.
4. It has been demonstrated [PARK90] that presenting a map to drivers while the car is moving will result in drivers taking their eyes off the road for unacceptable lengths of time. This result was obtained under very specific conditions¹ and may not apply universally.
5. It has been demonstrated [THOR80, STRE86] that some drivers do not understand maps at all. Some drivers have trouble understanding the spatial relationships presented on maps. This may be a matter of training.
6. There are good and bad positions to put displays and instruments. In some positions the driver can read displays more quickly and more accurately than in others. Generally the rule is directly in front of the driver and slightly below the sight line. The farther below the line of sight, the longer it takes the driver.
7. A HUD is effective if the point of focus is about four meters in front of the car. [Thom89].
8. It is believed that symbols are effective ways to communicate information. This seems to be a folk homily. Can we prove it?
9. Some messages must have higher priority than others. A "given." We make the assumption that a message warning of immediate danger must have higher priority than all others².
10. Sequences of messages are a critical problem. This normally occurs at or near difficult and complex intersections, traffic circles or highway interchanges. See for example [WEB90] or [CHAL90] where the problem is briefly discussed. In these cases, because the driver must make a sequence of manoeuvres in a brief period of time, it is necessary to communicate the complete sequence of instructions to the driver before the manoeuvres are started.

Some questions affecting integration arising from the principles given above. How do we structure these priorities? How many degrees of priority are there? What happens to priorities when a new system is added to a car? One might assume that the driver workload plays a role in the delivery of messages but not in the priorities of messages. Is this a valid assumption?

At this time, we are aware of no models of driver workload that are computable from indirect evidence, the input stream of message and sensors related to driver controls. It should not be surprising to realise that these same factors apply to the HMI aspects of other systems in a car. The aircraft industry³ has known for some time that too much information can overwhelm the pilot and actually cause a pilot to turn off or ignore information; even time-critical warnings!

It is our belief that the list of HMI factors above must be considered when developing any Navigation and Planning system. The list is not complete, nor is it verified. It represents current wisdom. As more research is completed we will have a better picture of these factors, and in many cases be able to quantify them precisely. In the meantime, the we recognise that they play a role in the design of our systems.

---

¹ More work is necessary to determine whether the same result would be obtained with a map presented on an digitised display placed where the driver could more easily focus attention on the map. Similarly the type of map displayed may be a factor. It is recognised by some groups implementing navigation systems (eg. CARMINAT, NISSAN) that a simplified version of a map may be better understood.
² Comments inserted after the workshop: As a result of discussions at this Näslingen Workshop on Navigation and Planning, we conclude that at least three levels of priority exist. These were tentatively classified by the time-related urgency of the message. Highest priority messages are those which must be reacted to in the shortest possible time, typically less than one second. An example of such messages are emergency warnings such as danger-of-collision or danger-of-hitting-a-pedestrian. The second priority group consists of messages that must be reacted to quickly, typically one second to 10 seconds. Suggested examples were a low-oil-pressure warning or a too-close-to-vehicle-ahead warning. The third priority group consisted of messages to which no immediate response is necessary. An example here were the ring of a telephone.
³ When the safety aspect alone is considered, it is not surprising that the aircraft industry has already made great strides in integration of the pilot-information systems in the aircraft cockpit. The Boeing 767 aircraft is one good example.
7. Summary

In this report we have argued that there are motivation, necessity and possibility for the integration of HMI aspects of the complex in-car systems emerging today. We have shown in particular that the HMI-Mgt function of Dialogue Management is possible.

Questions remain, of course. The study of message priorities must continue. A computationally tractable model of driver workload based on input related to message and control manipulation information should be incorporated. We feel that this could be included in a second generation HMI management model.

Acknowledgements

Annika Wärm contributed a great deal to this report through many hours of discussions and proof reading. Discussions with John Savelid and Anders Hallén in the Prometheus Sverige IRG project helped to clarify ideas. The interchange of ideas promoted by Stig Franzen with the Working Group 4 in Prometheus and the DRIVE groups working on HMI has helped to complete the picture for us.

References

Proceedings of the Prometheus Workshop III, Torino, 1990


[CREM90] M. Cremer Requirements from Prometheus Functions for Invehicle Architectures
Proceedings of the Fourth Prometheus Workshop, Compeigne, 1990

[FAIR90] S. Fairclough Comments on audible channel availability. Personal communication - to be published by HUSAT


[KLÆB90] R. Klæboe The Information Package and the Driver. to be published in

[TRON90]

[HUSA90] HUSAT Differences in driver behaviour due to the presentation of paper maps and Text display route guidance information. DRIVE VI017 Report 21


[SAND90] E. Sandewall System Block Specifications and their Prometheus Applications
Proceedings of the Fourth Prometheus Workshop, Compeigne 1990


11


[VERN90b] M. Vernet *Checklist of criteria to be used to evaluate ergonomics of incar information systems (DRAFT)* DRIVE STAMMI Workshop, München, March1990

Chapter 11

Planning the Route Guidance Dialogue

Annika Wærn
Planning the Route Guidance Dialogue

Annika Wärn
SICS, Knowledge Based System Laboratory, Box 1263, 164 28 Kista, Sweden
+46 8 752 15 00 (phone) +46 8 751 72 30 (fax)
annika@sics.se

Abstract
The general needs for reasoning when planning the presentation of route guidance information are discussed. We contrast the AI approach of situated planning to the more standard computer science approach of state transition networks, and discuss the advantages of each approach. The route guidance presentation task is first described as a situated planning problem, and then an alternative view is given using the state-transition model, which we argue is superior for this particular task. We finally show how one reasoning components necessary for constructing the route guidance dialogue is incorporated in the state-transition model.

1. Introduction

Interactive Route Guidance is common name for a set of driver aids that help a driver to find her way through a city. A route guidance system has two tasks that could be fulfilled separately: Firstly, it aids the driver to reach her destination by reacting dynamically to her actions while driving. Secondly, it has also the task of choosing a best route to get to the destination. Within Prometheus, the route guidance task is sometimes separated into two independent tasks solved by independent systems: "trip planning" and "route guidance". The trip planning is the task of planning a trip before it starts: This task involves selecting an "best" route with respect to the drivers wishes, and can require quite a lot of driver-system interaction. It could be envisioned as a separate program running at your home computer. "Route guidance" then is the task of aiding the driver, on route, in following the selected route. It must be realized in an in-car system.

In the SICS group effort, we have chosen to integrate both functionalities in the same system. There are several reasons for this: we believe that good on-road performance (in particular in selecting suitable messages to the driver) is dependent on what has been communicated during trip planning [Wä 90b]. But more importantly, in order to fulfill the task of choosing a best route the system must, apart from the driver preferences, also consider factors in the environment that might change over time, issues like weather, traffic congestions, road work, etc. Some of these can change at any moment, both before the trip starts and while the driver is on her way. Some reactivity, or replanning, is therefore needed in the system, and this is needed both during trip planning and during route guidance.

In related work [Wä 90a; HööKa 91] we have reported studies concerning the information needs for different types of drivers. The results show that the planning needed both for actually choosing a good route and for presenting it will be largely dependent on who the driver is. We need to reason not only about what she knows, but also about what she is currently intending, and what messages she is likely to capture and understand.

In this paper, we focus on the reactive nature of the driver-system dialogue. The system must react to changes in the current situation that the driver is not aware of, but it must also react and adapt to the driver's actions. In itself, it is only suggestive: it can inform the driver about suitable actions, but it never performs them by itself. The driver is an intelligent and independent actor, who sometimes will understand and perform the suggested guidance instructions, and sometimes will misunderstand an
instruction, or even ignore an instruction for various reasons. We must realize that even with the most advanced, adaptive system, the human driver will sometimes know more about the traffic situation at hand, and will therefore be a better judge of what to do, or will simply not accept to be the slave to the system.

The reactive nature of the driver-system dialogue has several impacts on the necessary reasoning component. The most notable effect is that the dialogue must be re-planned at times, the set of instructions that the driver will need cannot be foreseen in advance.

In AI planning literature the systems described are mostly one-agent systems (autonomous robot systems) [GeLa 87; AgCha 87; RoKæ 86], and only a few deals with multi-agent problems [La 88]. However, these approaches have the dynamic and real-time properties in common with the IRG task, which makes their solutions interesting to consider. On the other hand, the reactive planning problems are on the whole simpler in this domain than in the general planning domain, since the set of possible alternative solutions is always very small.

The paper is organized as follows:

In section 2, we describe two general approaches that both are applicable to our application. The first, situated planning, arises from the AI planning literature. The "state transition" view comes from classical computer science. We discuss how these approaches are interrelated.

In section 3, we use the HMI reasoning strategies as a target example for the two approaches, and discuss why the second is preferable for this particular application.

In section 4 we discuss how the reasoning needed is to be incorporated into the state-transition model described in section 3. We also briefly describe one particular reasoning problem that we have given a lot of attention at SICS. In [Li 91] we describe a demo system where the state-transition approach is taken as the basis, and where the particular difficulties discussed in section 4 are solved. In [Wæ 91], some other remaining reasoning problems are discussed and solutions proposed.

2. Situated Planning and State-transition networks: a comparison

2.1 Situated Planning - The AI answer to real time problems

The task of planning the actions of a system in a changing environment has been given a lot of attention in artificial intelligence. In the earliest planning approaches (see for example [FiNi 71]), planning is carried out as a complete search through a space of possible actions, trying to find a sequence or tree of actions that lead to the sought goal. This principle has later been criticized from two standpoints. The first stand-point is that it is very difficult to foresee all possible outcomes of an action to be able to cope with any situation that might arise, especially in a multi-agent world. The second stand-point is that typically, the search is very time consuming. For these reasons, this kind of planning is most suited for single-agent problems, if the time available for planning is not limited.

More recent approaches of the same type involve hierarchical planning [Wi 87]. In hierarchical planning, planning is seen as a task of breaking up higher level goals into sets of lower level goals, until a set of primitive actions is obtained. The set of primitive actions need to be ordered in time depending on how their outcome affect the possibility to perform other actions: some of them might need to be performed in a particular order. This gives the possibility to postpone some of the planning until more information is available: some higher level goals might be left un-expanded, or the set of primitive actions might be left partially ordered [Sa 77].

In the eighties, the planning community has concentrated on the real-world problem of reasoning in limited time. The classical planners all have the fundamental fault that they need to have completed their reasoning before they can act. In real time problems, we would like to construct a system that can make a quick, dumb solution if that is required, but still is able to reason intelligently if given more time. This is called the any-time property of a real-time system.
Georgeff and Lansky have constructed the Procedural Reasoning System PRS [GeLa 87], which can be viewed as a "committed" hierarchical planner with very advanced possibilities to act on new evidence. The theory of actions and processes that underlies PRS is described in [GeLa 85]. The PRS approach seems very suitable for planning problems of our kind, where the complete set of actions that might be needed is rather small and can be foreseen. For this reason, we choose to describe this system more in depth.

The system is viewed as an agent acting in a changing real-time environment. The system architecture contains a database (corresponding to the set of beliefs of an agent), a set of goals for the agent, a set of declarative procedures (Knowledge Areas or KAs) and an intention structure (corresponding to a set of KA's that the agent has decided to use). The KAs are represented as graphs, where the arcs are labelled with subgoals (including recursive goals), and can contain iterations and conditional selections.

In its simplest form, the interpretative cycle goes as follows: The agent has a certain set of goals and beliefs. These are matched against the invocation conditions for the KA's, and some KA's are selected for execution, and inserted into the intention structure. Next, the intention structure is examined to see what things can be immediately performed. These can either be atomic actions, or subgoals. If they are atomic actions, they are performed directly (affecting the external world). If they are subgoals, they are added to the goal stack. Some KAs (meta-KAs) can affect the system in other ways, such as manipulating the set of goals or the intention structure, guide the selection principle for goals from the intention stack, etc.

The PRS architecture allows for advanced real-time behaviour, such as adding new goals, depending only on new facts about the world (used for behaviours like to "duck" when a stone is thrown against you), and choosing a simple solution if there is not time enough to reflect on a better one.

Note that an essential philosophy underlying PRS is that the effects of an action cannot be presupposed to be true when the action has been performed. The actor's knowledge might have been incomplete or (more importantly) some other agent might have intervened. For this reason, some of the available KAs check whether certain goals have been obtained, and can be deleted from the set of current goals.

The main objection to the PRS system is that, in fact, the system does not do any planning in the classical sense. The set of possible KAs is selected by a simple unification against the goals and beliefs of the agent, there is no "looking ahead" to possible outcomes of actions at all.

One situated planner that actually does plan is the Entropy Reduction Engine ERE [Dr 89; BreDru 90] which contains a sequential planner, separate from the execution unit, that gives guidance to the latter. One weakness of this system (as originally described) was that the complexity of the planning task had not been reduced, and neither could the executing unit act on partial results from the planner. These problems are attacked in [DruBre 90] where hierarchical goal decomposition and probabilistic reasoning is used to guide the planner. This makes it possible for the planner to explore more probable alternatives first, and give advice to the "actor" as soon as a subgoal has been reached. The resulting planning algorithm is very intricate.

One can note that sequential planning lends itself easily to real-time implementations, since the planning starts from an initial world state and not from some goal. This world state might very well be the current situation. This property is not exploited in the work by Drummond and Bresina, but it is used in [Ko 90], where the any-time property is achieved by iterative deepening techniques, and each planning session starts from the current situation.

It has been argued that one of the reasons for not planning ahead is that humans very seldom do this [Su 87]. Conversely, it can be argued that this is unimportant since one in general want to use computer systems to improve performance in situations where humans encounter difficulties, such as situations where situated planning are needed. We do not take any general stand-point on this question. When we study our target application more closely, we will notice that at least in this domain, we will not benefit from a "planning ahead" possibility.
2.2 The "black box" view - The computer science answer to communication problems

A very common approach to software design is that of "divide and conquer". A system is designed and implemented by first giving a top-level decomposition into system parts defining their separate functionalities and interactions, and then iterating this decomposition into smaller and smaller parts.

An essential property of a "divide and conquer" approach is that the functionality of each of the separate parts can be described without describing the details of its implementation. This is called the "black-box" view of a system: We are not interested in the machinery inside the system part, the only thing we want to know is what response the system will give for a certain input. For "normal" programming modules, this is described as an input-output relation: We describe in some formal language what output the program should produce, given that the input is "correct". (See for example [LoSi 84].)

Programs that continuously accept input and provide output pose a particular problem. These cannot be described as a simple input-output relation. This kind of programs are discussed in particular in formal approaches to computer communication. One attempt to formalizing the black-box notion can be found in the CCS formalization [Mi 80;82]. In CCS, each communicating system is viewed as a finite state machine, where state transitions are associated with communication events. A system can be described in two different ways: by its finite state machine, and by its set of possible behaviours. In CCS one views two systems as equal if their set of possible behaviours is the same, even if their internal organization is different. Even if we do not want the same abstraction level as is given in CCS, it is useful to use finite automata to describe the behaviour of programs that continuously respond to input. We can describe the program as a set of state transitions, relating to these appropriate inputs and outputs. The exact behaviour in each state transition is in this case left unspecified (giving a "black-box" description at a lower level), specifying only what output is generated for what input in a certain state transition. We will subsequently call this the state transition model.

Another example of the "black-box" view in computer communication is the OSI partition of communication tasks into different communication layers. When defining the functionality for one layer, the OSI model only regards what kind of services the lower layer can provide, and how these can be accessed, but disregard completely how these services are realized within the lower layer.

These examples should make clear that a black-box view is very attractive if we want to describe communication between different systems or system parts. However, situated planning systems give a behaviour that is very difficult to describe as black-box behaviours. There are several reasons to this. One is that these systems try to attack an inherent problem to all "intelligent" systems: that of acting upon events that were unforeseen when the system was constructed. In PRS, this is mirrored by the fact that a goal is not assumed to be true just because the system has executed a KA to make it true; this has to be checked against the real world. In systems like the ERE this is mirrored by the fact that the system tries to plan ahead from the current situation, to find a suitable action. Another underlying reason to the complexity of situated planners is the aim to provide any-time properties (as described above, find a better solution if given time, but always be able to provide an answer.) This coupling between time elapsed and quality of result is very difficult to describe in a state-transition model.

The system that comes closest to the "black-box" view is the PRS system, since no real-time planning goes on in that system. Still, since the current world state is reviewed after each KA reduction, and a goal is not viewed as fulfilled until the system has checked it against the world state, the complete set

---

1 This problem has gained some attention in the situated planning community. Brooks [Bro 85] has suggested an alternative approach to constructing robot control systems, where the decomposition is task-oriented rather than functional. A control system is envisioned to consist of several autonomous subsystems, each geared to a particular task for the robot. Common functionalities would then need to be duplicated in the different subsystems instead of accomplished by specialized subsystems. A mixed task-oriented and functional approach is suggested in [K& 86], where the only functional decomposition is on the highest level, partitioning the system into a "perception" and an "action" component.
of state transition would be very large even for a simple PRS application\(^1\). For situated planners such as the ERE system, the task becomes almost hopeless, especially since the time allowed for planning in each step might be dependent on what is happening in the external world.

In [Bro 90a] we describe a functional decomposition of an IRG system into a route planner, the route monitor and the dynamic monitor, the meta-reasoner (or controller) and the HMI planner. The sixth party involved is the user of the system. This puts us in a rather awkward situation. On one hand, we deal with both system internal (module to module) communication and external (computer - user) interaction, which would suggest using a state transition approach. On the other hand, we are dealing with the user, which is an independent actor whose actions cannot be foreseen, which would suggest a situated planning approach. In the next section, we will try to describe the HMI reasoning strategies from both these views. The end result is that a black-box suffices.

3. IRG HMI reasoning as an example

3.1 IRG dialogue as a situated planning problem

The overall goal for an entire IRG system and the driver (together) is to reach the destination for the trip. To achieve this goal, the IRG system has an overall plan, consisting of two sequential goals:

1. Perform trip planning, and then
2. Give route guidance.

During Trip planning the HMI reasoner must fulfil two goals:
- Collect a complete and correct set of driver preferences.
- Give the driver an appropriate amount of pre-trip information about a route choice.

Route Guidance consist of fulfilling only one goal, that of reaching the destination. However, the system need to maintain three requirements, that all affect the behaviour of the HMI reasoner:
- The driver must have enough information to follow the chosen route.
- The current position should be on the chosen route.
- The current route choice should be "best" with respect to the driver's preferences and the current external information (which both may change).

To perform trip planning, each of the HMI subgoals are realized by a corresponding subplan:
- Collect the set of driver preferences: Some of them will be gotten by driver input, (such as the goal for the trip), whereas others will be given by using knowledge about the driver type, or the specific driver.

The set of preferences might be changed, if the driver does not accept a presented route. These changes might be based on further input from the driver, or on default guesses about what might be wrong.
- When a first guess on the driver preferences has been constructed, the route planner can be asked for a route plan. If it is possible, the entire generated route is described to the driver in a comprehensive form. The dialogue between the driver and the system can be more or less complex, varying from completely one-way, where the system just presents an entire route description, to

\(^1\) The situation becomes even more complex as the constructors of PRS have taken pain to allow Meta-KAs to take effect at any time in the interpretation cycle.
advanced two-way, where the presentation is given piece by piece, and the driver is allowed to acknowledge, reject, or question any part of the route choice.

If the driver indicates that she does not understand the route, or that she wants further clarification of it, some additional information must presented to the driver.

\[\text{Trip planning} \]

\[\text{Gather preferences} \rightarrow \text{Present route} \]

\[\text{Resulting route OK?} \]

\[\text{Yes} \rightarrow \text{Preferences correct} \]

\[\text{No} \rightarrow \text{Presentation understood?} \]

\[\text{Yes} \rightarrow \text{Initial presentation complete} \]

\[\text{No} \rightarrow \text{Trip planning complete} \]

\[\text{Figure 2. The subgoals in the trip planning phase. The dotted arrows represent subgoal completion, the unbroken arrows represent time dependencies.}\]

For the route guidance phase, the HMI reasoner need to act when one ore several of the system maintenance goals have become false. The goal for these are to re-establish the maintenance goals to be true.

1) If the driver does not have enough information to chose the chosen route, additional information must be presented.

This includes presentation of such information about the route that could not be presented during the trip planning phase, or repetition of information from the trip planning phase when necessary.

If a new route plan has been selected (or maybe even while it is being formed), the driver sometimes need to know about it. In particular, the driver must be informed if the new plan contradicts some information given earlier.

2) If the current position deviates from the intended route, the most complicated situation for the HMI planner arises. In this situation, the system must first try to determine whether the deviation was intentional or unintentional. If it was done on purpose, the driver’s preferences were understood wrong or have changed. If it was unintentional, the route simply needs to be replanned to find the best way out of the situation.

3) For the third subgoal, only the driver’s preferences concern the HMI planner. We can detect that the driver’s preferences have changed either if she tells us so, from an intentional deviation, or from
the driver rejecting a new route description. The possibilities for driver-system interaction is very limited during route guidance, why preference changes essentially must be guesses, based on what route the driver actually is driving.

We can immediately note some details that makes it unattractive to implement the HMI reasoning as situated planning. Firstly, there is very little need for "looking ahead" in each situation, which makes approaches like ERE unattractive. It is possible to envision a solution where the choice of presentation is based on how the most effective dialogue should look like. This presentation could be achieved by "looking ahead" for possible answers from the driver, and the subsequent dialogue. However, since the possible driver input is very restricted, there is hardly any reason to do this planning "on the spot", the correct choice can be precompiled.

Secondly, we should note that the time for planning what to do seldom is the crucial time restriction: instead, it is the time required for driver-system interaction that is crucial. Some dialogues can only take place when the driver is under low stress load, and some messages require larger head warnings than others. This implies that we could choose to first plan a message, and then replan if we discover that it could not be delivered. This makes the need for any-time properties very small.

Finally, the most difficult problem for the HMI planner is that several of the goals require knowledge about the driver's knowledge and preferences. This knowledge will never be precise: the system can only guess at it. The PRS solution of determining whether a goal is fulfilled by checking against the external world is here useless, since we cannot tell whether these goals are fulfilled that way.

Instead, a common situation will be that we assume that a property holds, only to discover later that it does not. This causes the activation of some subplan. In particular, the subplans corresponding to maintenance goals are not activated directly from the need of fulfilling one of the main goals, but because some event caused the system to understand that a property need to be reconsidered. To implement this in PRS, we would need a bunch of anonymous KA's, triggered from the current situation and not from the set of current goals. This is perfectly feasible, but goes against the original intentions of the PRS system in that there is hardly any hierarchical structure left at all.

There is still one possibility for limited situated planning: It is possible that the hmi reasoner could plan ahead while the system is waiting for the driver to respond. In particular, the system could look for what to do next if the driver rejects or questions a message. However, this will frequently cause the hmi reasoner to trigger the route planner to plan ahead "just in case" the driver will reject or question the route choice. It is very unlikely that this will give a good system behaviour, in particular if the hmi reasoner can plan an unlimited number of communication steps in advance. If we allow only a limited number of communication steps to be planned in advance, the behaviour can be described in using a state-transition model as well as using a situated planning scheme. Note also that the set of possible system outputs cannot be increased by such advance planning.

### 3.2. A state-transition model of IRG dialogue

One particular property that we can note from the previous section is that the system actions often are triggered by some input: The subplans corresponding to maintenance goals are not executed because they are part of a larger plan, but because the current situation causes them to be necessary to execute.

From the description of the IRG system architecture [Bro 90a] we can add the observation that the entire set of possible input to the HMI module is limited. This is true even for the set of possible messages from the user, due to the limited communication channels and the limited amount of attention that the driver can give to the system.

Finally, we have noted that the property that the time restrictions put on reasoning are not serious, and that the most important time requirements are instead put on the human-computer interaction.

Starting from these limitations, it is possible to describe the HMI module as a state transition system. Each arch correspond to one input to the HMI module and a related output from the module: all reasoning that needs to take place to construct that output is invisible, and for the purposes of this description, viewed as instantaneous. The states correspond to different types of waiting situations for
the HMI module. The HMI module does not do any reasoning while in a waiting state (else, we might have needed tacit state transitions). In some situations, the HMI is waiting for a particular input, but the fact that time passes while the module is waiting makes it necessary to allow the HMI reasoner to accept all possible inputs in each waiting state\(^1\). In [Wæ 91] we will discuss the relationship between the goal-oriented description of the HMI reasoner and this state-based description more formally.

The complete set of inputs to the HMI reasoner are:
Input 1: The driver indicates that she wants to use the system.
Input 2: The route planner has generated a route.
Input 3: The driver indicates that she understands and accepts a presented route.
Input 4: The driver indicates that she does not understand a presented route.
Input 5: The driver indicates that she does not accept a presented route.
Input 6: The route monitor signals that it is time to give some particular message to the driver.
Input 7: The route monitor signals that we have gone off route.
Input 8: The meta-planner signals that the driving has started.
Input 9: The meta-planner signals that we have reached the destination.

---

\(^1\) Some input events are sometimes impossible, depending on the functionality of the other IRG modules. For example, the route monitor will not send a message that it is time to present a route fragment to the driver if it already has signalled the we have gone off route.
Figure 3. A state-transition graph describing the system functionality

Depending on what state the HMI reasoner is in, the action triggered will be different. In the state transition graph in Fig. 3, we can see how different events cause different state transitions. The reasoning and output actions needed in different state transitions are described below.

For the case of convenience, we assume that input 1, the startup, only happens at the beginning of the session (state transition 1 -> 2). We need to create the set of driver preferences (which might require some communication with the driver) and then trigger the route planner to plan a route given these preferences.
Input 2, a new route plan from the route planner, can be generated both during the trip planning and the route guidance phase, and for several different reasons. During trip planning we have three cases:

- If it is the first route presented to the driver ever (state transition 2 -> 3), we want to describe the entire route in some comprehensive form to the driver if this is possible, and check that the driver understands and accepts it.
- If it is an alteration to an old route plan (state transition 3 -> 3 and 5 -> 3), we want to present the new route in a way that highlights the differences to the old route choice.
- If the new plan was generated instead of one that the driver did not understand or understood only partly, (state transition 4 -> 3) we just tell the driver to disregard the previous route description and present the new one.

In on-route mode, we have two cases:

- If the route was generated while the driver is on route (state transition 6 -> 7), we must first decide whether the driver should be notified about the changes. If she should, we try to generate a comprehensive description of the changes of the route plan, and present this immediately to the driver. We might in this case allow the driver to reject the new plan. If it is not possible to make a comprehensive description, we just indicate that the route plan has changed, and give the information piece by piece at appropriate positions.
- If the route was generated while the driver was off route (state transition 9 -> 8), the new route must be presented. The presentation is done as above.

Input 3, a message from the driver accepting a presented route, can occur only when a route has been presented. During trip planning this occurs in state transitions 3 -> 5 and 4->5, and during route guidance in state transitions 7->6 and 8->6. In both cases we need to send a message to the planner that the route choice was accepted by the driver, and prepare future on-road information.

Input 4, a message from the driver that she does not understand a route description, is treated differently during trip planning and route guidance. During trip planning (state transition 3 -> 4) we first try to find an alternative description. If this is not possible, we simply postpone presentation until the route guidance phase. During route guidance (state transition 7-> 6, state transition 8 ->6), we cannot afford the luxury of making a new comprehensive presentation. Instead, we just postpone presentation and give it piece by piece at appropriate positions.

Input 5, a message from the driver that she rejects the chosen route, is also treated differently in different situations. During trip planning (state transition 3 -> 3 and 4 -> 3) we try to change the set of driver preferences so that they better suit the driver's wishes, and then replan according to the changes. If the driver preferences could not be updated, we just replan to get "the next best" route choice from the same preferences. During route guidance (state transition 7 -> 6 and 8 -> 9), we can also try to make changes to the set of driver preferences, if we can do such changes without communicating with the driver, and then trigger replanning. If we are on route with respect to an old plan (state transition 7 -> 6), we can also choose to keep the old plan instead of triggering replanning.

Input 6, that when the route monitor indicates that we have reached a point where a message should be delivered to the driver, occurs only while the driver is on route during route guidance, and causes no state transition (6 -> 6). We need to send a message request to the dialogue manager, and to decide the next suitable point for presentation and send a request to the route monitor to indicate when this point is reached.
Input 7, that when the route monitor indicates that the driver has deviated from the intended route, occurs only in state 6 (during route guidance) and causes a state transition 6 \rightarrow 9. This situation is the most difficult one to handle. We need to:

- Determine whether the deviation was done by accident or on purpose.
- If it was by accident, we must first inform the driver about the fact that she has deviated. We can also try to determine what was wrong in the model of the driver knowledge. If the route monitor has not already triggered replanning, the hmi might need to do it.
- If the deviation was done on purpose, we try to make appropriate changes to the set of driver preferences, based on the route the driver actually is driving. When this is done, we should trigger replanning to generate a route plan according to the changed set of driver preferences.

4. Reasoning mechanisms needed

From the description in the previous section, we can see that although we have given the HMI reasoner a high-level description that is static and determinate, all details about how specific problems are solved are left unspecified. Within some state transitions, we need to employ different reasoning methods to solve the more complex tasks of the HMI module. In [Wea 91] we will examine some such reasoning problems, and how these could be attacked. Here, I will just briefly discuss one such reasoning task, that we have chosen to focus upon at SICS.

From the previous section, we can see that the system frequently needs to present a new route plan to the driver. This route plan is in most cases very likely to be similar to the ones already presented. It might at a first glance seem like a simple problem to produce such a description, but careful examination reveal some difficulties. The SICS research group has put a lot of attention to the problem of presenting routes and route changes. In [Li 91] we will report a demonstration system that is capable of presenting route changes. The system incorporates a route chunking mechanism [HöökKa 91], a presentation planner based on the principles described in this paper, a route planner [Bro 90b; GuLi 91; BroLi 91] and a presentation module transforming messages into natural language.

As the first difficulty, we must note that we must describe the new the new plan by relating it, not to the old route plan, but to the old route description, as that is the "common denominator" for the system's and the users' knowledge. Intuitively, this is due to that when the system describes the route to the driver, it does not transfer the route itself but a description of it. However, it is not convenient to store the description in the exact output syntax, for one reason because that syntax is chosen by the dialogue manager. Rather, we would like some intermediate format that is expressive enough to describe solely the information content of the messages.

The second difficulty arises is that the new route might still fit the old description. There are two reasons for this: The initial description contained "gaps" where the driver was assumed to understand the route by himself, or the system might have postponed the presentation of some parts of the route until later (like right before they are traversed). Thus, the system must consider what to do when the new route still fits the old description. The exact action in this case depends on how other parts of the system, such as the route chunking algorithm, are constructed. For our purposes, it suffices to note that the system must be able to deduce that a new route fits an old description. For this reason, we must carefully define the semantics of the intermediate route description format.

The third difficulty is that the current route choice might have been transferred to the driver in several messages. For example, only parts of the route may have been described. Secondly, the route might have been replanned before, and consequentially, the driver might already have been told about changes from the original description. Thus, the new description must be related to the whole sequence of plan descriptions, and not only to the original, or the most recent one.

This property is actually true for all messages passing from the HMI module to the driver, and it suggests that the what module should contain a dialogue memory that mirrors the driver-system
dialogue over time. Based on the state-transition model, we can model time as event histories. Each event is then characterized by the incoming message and the current state, and the resulting outgoing message(s). An event is not in the purest sense atomic, since the reasoning involved will take some time, but it is atomic in the sense that no new incoming message can abort it.

5. Conclusions

We have shown by example that a simple state transition model might be a good basis for design and implementation, even if the reasoning tasks involved might be rather complex, and situated replanning is needed. We have discussed under what circumstances this is possible: those of having a limited set of possible inputs, and low (but not nonexistent) requirements on real-time behaviour. We have also argued that such an approach is preferable if it is possible due to the possibility for modular definitions.

6. References


