Draft Version
Published in International Journal of Intelligent Systems, vol. 25, no. 7, 2010

Interactive In-Vehicle Guidance through a Multi-Hierarchical Representation of Urban Maps

C. Galindo, J. Gonzalez, J.A. Fernádez-Madrigal
System Engineering and Automation Department
University of Malaga, Spain
e-mails: {cipriano,jgonzalez,jafma}@ctima.uma.es

Abstract

Small computers used for assisting drivers (mostly in finding routes) have been growing in popularity in the last years. These systems are inherently interactive, but up to date this interaction is tackled under rather simple approaches. For example, current routing computered-assistants only consider the shortest or the quickest route to a destination, although in certain situations it could be interesting for the driver to take into consideration other factors, such as the criminal rate, land value, or the beauty of the areas to be traversed. On the other hand, the interactive processes between the driver and the routing assistant are still very limited: they only enable the user to discard (or suggest) particular locations through a fixed set of names, i.e. street's names. This paper proposes a novel interactive mechanism for in-vehicle routing that uses topological information at different levels of detail and a multi-hierarchical representation of urban maps. These hierarchical representations permit the system not only to plan routes efficiently, but also to report them at different levels of detail in a human-like set of symbols adapted to each user. This enhances the human-computer interaction during the routing process, increasing driver satisfaction. We illustrate our technique through a case of study in the city of Málaga (Spain).

Keywords: Human-Computer interaction, Interactive routing, Multi-hierarchical representation of space

This work was supported by the Spanish Government under Research Contract CICYT-DPI05-01391.

1 Introduction

Advanced Traveller Information Systems (ATIS's) are an emergent technology that copes with intelligent transportation applications aimed to provide vital traffic information to users. ATIS are an integral component of the concept of Intelligent Transportation Systems (ITS) that provide solutions to improve the transport of goods and people (ISO, 1998). ATIS's users can range from traffic control centres (that can manage, for instance, traffic information to coordinate emergency services or a truck fleet) to drivers in general that can request information in order to plan their trips. Especially in this last case, interaction between the ATIS and the user is of a paramount importance.

Due to the characteristics of real road networks, ATIS's have to access ITS applications to gather enormous amounts of information regarding road topology, traffic congestion, state of the pavement, relevant facilities, etc. For those purposes, geographical information systems (GIS) have become a popular and useful tool for storing, analyzing, and visualizing geographical data. Within GIS-based ATIS applications, an important component is the *route planner*, which provides the user with the best path to arrive at a given destination in terms of time, travelled distance, or any other relevant measure. Nevertheless, in general, these pre-programmed criteria may not be the most convenient or may not lead to a complete user satisfaction in certain situations. For example, a driver might prefer to pass through a particular neighbourhood, albeit such a decision increases the travelling time. Also drivers may have reasons to avoid certain areas (for instance, because they dislike the aspect of their buildings) even when passing through them is necessary for achieving the quickest route. These user motivations can not be easily considered by conventional routing algorithms unless the driver actively participates into the route planning process.

This problem is known as *route choice* and has been addressed actively in the literature (Miller and Shin-Lung, 2001). Different models have been proposed to try to capture the driver behaviour when choosing routes, ranging from deterministic ones that provide routes with respect to prefixed criteria (Azevedo et al. 1993, Ben-Akiva et al. 1984) to stochastic models which generate random routes based on utility functions

(Bierlaire et al. 1995, Nielsen et al. 2002, Bolduc and Ben-Akiva 1991). However, all of them may fail at proposing routes for specific drivers, since her/his behaviour is highly tight to psychological and emotional aspects that can be neither measured nor forecasted by any system.

In this work we rely on active user participation for addressing the route choice problem considering large amounts of data, as it is the case of real urban networks. Our approach considers topological information constructed on geographical data at different levels of *abstraction*, which stems from grouping elemental data into more general ones, i.e., districts are the result of grouping a set of neighbourhoods, which in turn result from grouping streets, junctions, buildings, etc. The use of abstraction in this way is widespread and has been proven to be certainly useful for reducing the computational effort of path finding, even making tractable some problems that otherwise are not (Galindo et al., 2004). In addition, we enable the driver to construct his own hierarchy of spatial concepts resembling his own map-in-the-head, as stated by psychological research (Patalano et al. 2001, McNamara et al. 1989, Holding 1994). This provides the user with a hierarchy of abstract concepts suitable for machine-driver communication[1]. For simultaneously coping with efficient route planning and route communication-interaction, we have implemented a multi-hierarchical structure, called *Multi-AH-graph*, that has proven its suitability for task planning in the robotic arena (Fernandez et al. 2002, Galindo et al. 2004). Within the transportation and cartographic domains, the use of hierarchical models (also known as multi-scale or multi-resolution models) is widespread, but uniquely devoted to improve routing or map visualization (Mackaness et al. 2007, Jagadeesh et al. 2002, Queck and Srikanthan, 2001, Smith and McGinty 2001, Timpf and Frank, 1997, Huang et al. 1997). Up to the best of our knowledge, no work has considered yet multiple hierarchies for interactive routing.

The main features/advantages of our interactive routing system are:

a) The user is initially provided with the best path (following any pre-programmed criteria, e.g. the shortest or quickest path) at different levels of detail, involving topological information and labels created by the driver. For instance, at a high

[1] In this work we do not cope with the human-system interface, which could be textual, voice-based, visual, etc. Our approach fits well to any of them apart from the technical issues to be solved for each case, e.g. voice recognition within a noisy situation for a voice-based system.

level of abstraction, the driver can obtain the following route to go to a particular street of “the shopping area”: Start from “My work area”, pass by “John’s neighbourhood” and then arrives to “the shopping area”. Thus, the user gets a route that has special meaning for him. Through this abstract route, the driver may wish to avoid, for instance, the John’s neighbourhood for some particular reason, and thus, s/he requires an alternative route, let’s say passing by “the train station area”. Discarding abstract information permits the routing algorithm to get rid of all the elements contained into the discarded place (in this case, all the streets, junctions, etc. entailed by the John’s neighbourhood), and thus to improve the efficiency in the path searching process (Fernandez and Gonzalez, 2002). It is important to remark that the hierarchical topology used for route communication is manually constructed by the user to reflect his own view of the map through his particular arrangement of information and locations’ labels.

b) A driver familiarized with the roadmap of a city may accept an abstract path when s/he is well oriented. For instance, a taxi driver may accept a route given in terms of areas since s/he does not need further information for traversing them. In this case, the user is provided with a prompt and useful solution rather than with a highly detailed route whose calculation and communication to the user would take more time.

In the ATIS literature the use of single hierarchies for improving route planning has been widely exploited (Jagadeesh et al. 2002, Quek and Srikanthan 2001, Huan and Jing 1996, Car and Frank 1993). In this work we consider a two-hierarchical structure in which one hierarchy is engaged with efficient route planning, while the other serves to translate the resultant routes to a human-friendly language, enabling the driver adaptation and interaction with the system.

The structure of the paper is as follows. Section 2 presents some related works. Section 3 describes and formalizes the multi-hierarchical topological map (MAH-graph) used to represent urban maps. Section 4 presents the proposed interactive route planning. In section 5, our method is illustrated in the particular case of the city of Málaga. Finally some conclusions and future work are outlined.

2 Related Works

The problem of calculating suitable routes for a *particular* driver is an interesting, relevant issue for in-vehicle guidance systems. This problem can be formulated as how a computational system can choose the most convenient route out of the huge number of solutions. A simple solution could be to use a deterministic route generation model to yield those routes that minimize a certain criteria, e.g. distance, travelling time, etc, but obviously, the resultant routes may not always convince the drivers. An alternative consists of generating a set of possible routes by modifying an initial one, i.e. the shortest route, by adding or removing links (Scott et al. 1997, Azevedo et al. 1993, Ben-Akiva et al. 1984). Among the set, the driver could select the one which fulfils her/his requirements. But such a set may not contain yet the route desired by the driver.

There are many works addressing the drivers' behaviour when deciding routes (Ross et al. 1997, Mannering et al. 1995, Dingus and Hulse 1993, Khattak et al. 1993), though their results have not helped for improving choice route models at a large extent. In an attempt to capture the behaviour of drivers when choosing routes, different stochastic models have been presented to generate wider and more diverse route sets based on utility functions and random variables (Bierlaire and Frejinger 2005, Cascetta and Papola 2001, Bolduc D. and Ben-Akiva 1991). However, and given that user motivations to decide a route depend on multiple and, in most cases, subjective factors (e.g., driver mood, past experiences, appealing to pass through or to avoid a particular area at a certain moment, etc.), even stochastic models can not capture faithfully the driver behaviour. In the last years, in-vehicle routing systems coping with deterministic models, such as *TomTom*, have become very popular. Although these solutions exhibit acceptable results for finding the quickest or shortest path to a destination and certain interactive capabilities, they have a common characteristic: the system only reports low-level information to the driver in terms of pre-built street's names, and as commented, the pre-programmed criterion may largely differ from the driver interests when choosing a route, i.e. passing through beautiful but crowded areas.

We believe that an active human participation into the routing process, and consequently a proper human-system communication, is the key for providing the driver with the best route with respect to her/his desires. Our work shares some similarities with the one reported in (Scott et al., 1997). It proposes *alternative paths* as modifications of the best path given in terms of low-level information (streets, junctions, roads, etc.), where the user can select certain edges (streets or roads) of the topology as forbidden elements which must not be utilized to calculate the resultant route. This forces the calculation of an alternative path (this feature is also supported by commercial tools). The main limitation of this solution is that, in general, the driver does not have a comprehensive view of the whole route, and/or the information is only given in terms of streets.

In our approach, a complete, high-level route involving understandable and personalized topological information is given to the driver. This high-level route is progressively refined until a detailed one is found, allowing the driver to be completely informed about the routing process and to interact (e.g. adding/removing places/links to be traversed or avoided) at any level if needed, without compromising the efficiency of the routing algorithm.

3 A Multi-Hierarchical Representation of Urban Maps

In this section a Multi-Hierarchical topological/geographical representation for urban maps is presented. First, a single-hierarchy representation is described, and then, section 3.2 details its multi-hierarchical extension. Both representations are based on a generic hierarchical graph-based model which is mathematically formalized in section 3.3 for completeness. The application of this structure to model urban maps is one of the contributions of this work since previous related proposals do not have the generality of ours (Miller and Shin-Lung 2001, Timpf and Frank 1997).

3.1 Hierarchical Representation of Urban Maps

Urban maps and, in general, geographical data are usually represented in a layer-based fashion, through multi-resolution or multi-scale spatial databases in which each layer represents the same spatial data at different levels of resolution (Mackaness et al. 2007, Paiva et al. 2004, Jones and Zhou 2001, Miller and Shin-Lung 1997, Jong-Hyun et al 1999, Timpf and Frank, 1997).

Following this convention, we represent topological information (e.g. streets, junctions, and groups of them) stemmed from urban maps through hierarchical graph-based structures called *AH-graphs* (Fernandez and Gonzalez, 2002). An AH-graph is a graph representation of topological information which includes the possibility of grouping pieces of information into more *abstract information* (for instance, a set of streets and junctions into a neighbourhood). This kind of abstraction produces different layers isolated from one another, called *hierarchical levels*. Hierarchical levels are multigraphs[2] whose nodes can represent relevant locations of the topology (i.e., junctions, buildings, or abstract areas), while edges indicate their connectivity, i.e., streets, avenues, roads, etc. In an AH-graph, a group of nodes of a hierarchical level can be abstracted (that is, grouped) into a single node at the next higher level, which becomes their *supernode* (the original nodes are called *subnodes* of that *supernode*). Analogously, a group of edges of a hierarchical level can be represented by a single edge (their *superedge*) at the next higher level.

The lowest hierarchical level of an AH-graph, called the *ground level*, represents urban maps with the maximum amount of detail available, i.e. in terms of streets and junctions. The highest hierarchical level is called the *universal level*, and it typically contains a single node that, in our case, represents the whole city. The number of hierarchical levels of the AH-graph depends on different factors. For a hierarchy used for route communication the number of hierarchical levels depends on the particular arrangement made by the driver. For instance, figure 1 shows an example of an urban map modelled by an AH-graph in which the driver has considered five levels of detail (streets → neighbourhoods → districts → areas → city) that fulfils her/his requirements for being informed about urban routes. It is important to highlight that the clustering is performed by the user and thus, for instance, the neighbourhood labelled by the user as "John's neigh." comes from a particular arrangement of streets and junctions that may not correspond to the administrative meaning of the correspondent area.

[2] A *multigraph* is a graph with multiple parallel edges, i.e. edges that have the same end nodes. We also use the term "graph" to refer to multigraphs.

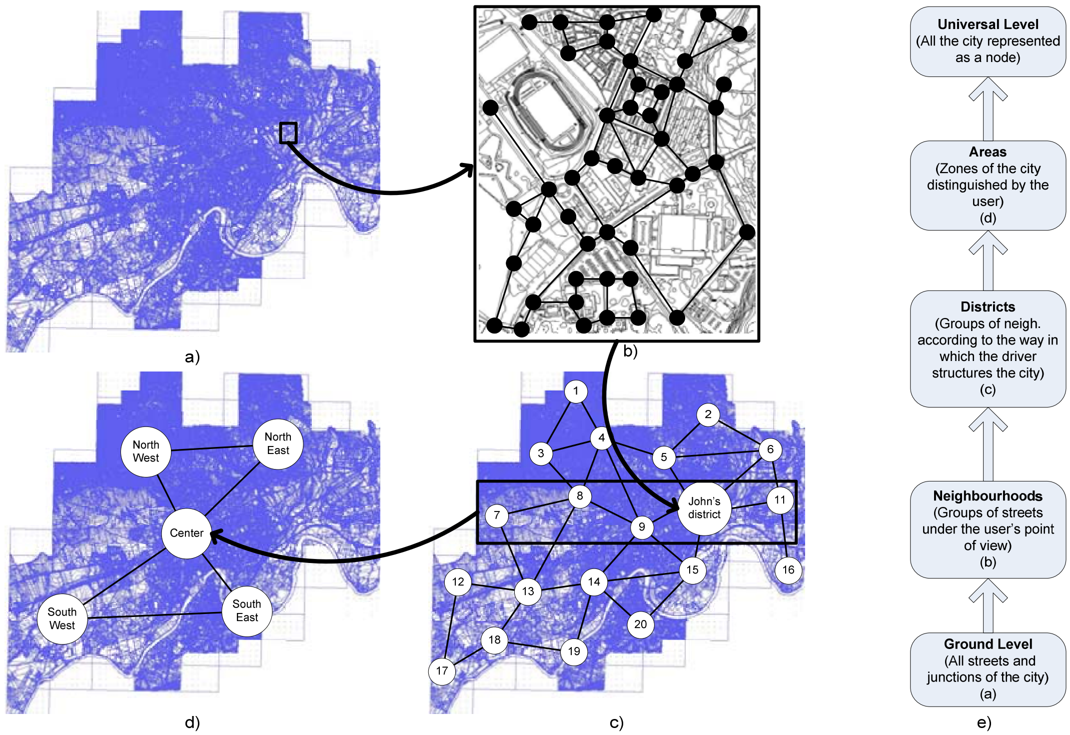

Figure 1. An AH-graph example for modelling an urban map. a) A graph representation of the topology of a city. b) A zoomed portion of the city that represents a neighbourhood (labelled by the driver as "John's neigh."for instance. c) Nodes represent groups of neighbourhoods, i.e. districts with a representative label for the user. d) Districts are grouped into areas following the user criteria. Chart e) shows a scheme of the different levels of the AH-graph, from the ground level (lowest level) up to the universal level which represents the whole city with a single node.

Besides the topological information captured by the AH-graph at any hierarchical level through nodes and edges, they can also hold pieces of non-topological information in the form of *annotations*. Annotations can be seen as pieces of thematic information that may include, but is not limited to: geometrical data (i.e., distance or area), costs incurred in traversing edges in time or consumed fuel, administrative data (i.e., land value, criminality rate, etc.), tourist information (e.g., number of monuments), etc.

As commented, the construction of the hierarchies (AH-graphs) used for route communication/interaction must involved the driver participation to indicate her/his particular groupings and labels. However, when using AH-graphs for other purposes, e.g. planning routes, automatic processes can be developed for the automatic generation/maintenance of hierarchies. We do not cope with such processes in this paper since it has been presented elsewhere (Fernandez and Gonzalez 2002, Galindo et al. 2007).

3.2 A Multi-Hierarchical Representation of Urban Maps

A single hierarchical representation (or AH-graph) is the basis for constructing a more elaborated representation of urban maps through *Multi-AH-graph*s structures. Broadly speaking, a Multi-AH-graph is a set of hierarchies (AH-graphs) interwoven in a directed acyclic graph whose nodes correspond to hierarchical levels (see figure 2). This type of representation based on multiple hierarchies yields two important benefits with respect to single-hierarchical models:

- When multiple hierarchies are available tuned to efficiently perform path planning under different criteria, a routing algorithm can choose the best one to solve the problem at hand.

- Resultant routes can be expressed by using information from any of the hierarchies, and thus the result can be given in the most suitable way for each specific purpose and user. Normally, different drivers may consider different groups of topological information and/or with different names (for instance a certain district of a city can be labelled as “the shopping area” by a customer, but as “my work place” by a shopkeeper), and thus the routing algorithm should provide each user with the computed routes in terms of her/his particular ontology.

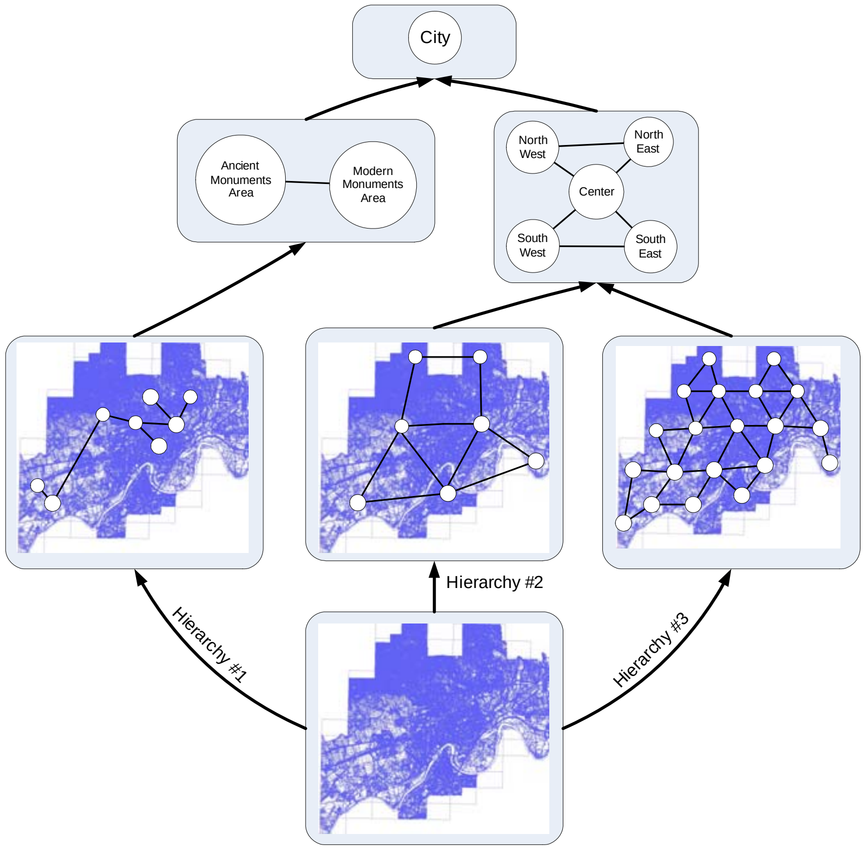

Figure 2. A Multi-AH-graph example with three hierarchies. Upon the common ground level a number of hierarchies can be manually constructed to satisfy different users or to cope with different situations. For instance, in this figure, hierarchy #1 divides the city in two parts according to the type of monuments they contain (useful for tourist drivers); hierarchy #2 represents the distribution and coverage of police stations, and hierarchy #3 corresponds to the AH-graph of figure 1 that represents the city division in administrative areas (i.e., zip codes).

The use of multiple hierarchies for representing space is not widespread in the literature. It has been addressed in (Galindo et al. 2006, Thomas and Donikian 2003), though this is the first time it has been proposed for modelling urban maps and for interactive routing.

3.3 A Mathematical Model for Multi-Hierarchical Representation of Urban Maps

For completeness, this section mathematically formalizes the *Multi-AH-graph model* (previously presented in (Fernandez and Gonzalez 2002)) from which single and multiple hierarchical representations of urban maps can be derived. Firstly we give a formalization of flat graphs.

3.3.1 Flat Graphs

Flat graphs considered in this work are non-empty, finite, annotated and directed multigraphs, *G*, defined for convenience as $G = (V, E, \gamma, ini, ter, \alpha_v, \alpha_e)$, where *V* is the finite, non-empty set of *nodes*, *E* the finite set of *edges*, γ the *incidence function*, *ini* the *initial function*, *ter* the *terminal function*. α_v and α_e are the *annotation function for nodes*, and for *edges* respectively which are not considered in this work. The following requirements must be satisfied:

$$V \bigcap E = \phi, \quad V \neq \phi \qquad (1)$$

$$\gamma : E \rightarrow (V \times V - \{(a,a) : a \in V\}) \qquad (2)$$

$$ini : E \rightarrow V; ter : E \rightarrow V \text{, such that } \gamma(z) = (ini(z), ter(z)) \qquad (3)$$

The incidence function $\gamma(z)$ yields the nodes connected by the edge *z*. By constraint (2), there may be more than one edge connecting the same pair of nodes in any direction (*parallel* edges), but no edge can exist connecting a node with itself. In this paper, *V* will be referred to as $V^{(G)}$, *E* as $E^{(G)}$, and γ, *ini*, *ter*, α_v, and α_e as $\gamma^{(G)}$, $ini^{(G)}$, $ter^{(G)}$, $\alpha_e^{(G)}$, $\alpha_e^{(G)}$ respectively, as long as the graph to which they belong must be specified explicitly. The set of all non-empty, finite, annotated, directed multigraphs will be denoted by Θ, and they will be referred from now on simply as *graphs*.

3.3.2 The Multi-AH-graph Model

A Multi-AH-graph can be formalized as a set of graphs and abstractions between them (see section 4.1 for a descriptive example). An *abstraction from graph G to graph H* (see figure 3) is a partial homomorphism[3] between both graphs, defined as a tuple, $A = (G, H, \upsilon, \varepsilon)$, where *G* is the graph that is abstracted, *H* is the resulting graph, υ is the *abstraction function for nodes*, ε is the *abstraction function for edges*. Other classical mappings between graphs can be considered. In our Multi-AH-graph model, the following restrictions must hold[4]:

---

[3] A graph homomorphism is a mapping between two graphs, *G* and *H*, that respects their structure, that is, a function $h : V^{(G)} \rightarrow V^{(H)}$ such that $\forall z \in E^{(G)}, \exists x \in E^{(H)} : ini^{(H)}(x) = h(ini^{(G)}(z)) \wedge ter^{(H)}(x) = h(ter^{(G)}(z))$. A partial morphism is a morphism between two graphs whose domain is not the complete first graph, i.e., it is not defined for every node of the first graph.

[4] We denote with *def(g(x))* the fact that *g(x)* is defined, i.e., that element *x* belongs to the domain of function *g*.

$$\upsilon : V^{(G)} \rightarrow V^{(H)} \text{ is a partial function.} \quad (4)$$

$$\varepsilon : E^{(G)} \rightarrow E^{(H)} \text{ is a partial function.} \quad (5)$$

$$\forall z \in E^{(G)}, def(\varepsilon(z)) \Rightarrow [def(\upsilon(ini^{(G)}(z)) \wedge def(\upsilon(ter^{(G)}(z))] \quad (6)$$

$$\forall z \in E^{(G)}, def(\varepsilon(z)) \Rightarrow \upsilon(ini^{(G)}(z)) \neq \upsilon(ter^{(G)}(z)) \quad (7)$$

$$\forall z \in E^{(G)}, def(\varepsilon(z)) \Rightarrow \begin{bmatrix} \upsilon(ini^{(G)}(z)) = ini^{(H)}(\varepsilon(z)) \wedge \\ \upsilon(ter^{(G)}(z)) = ter^{(H)}(\varepsilon(z)) \wedge \end{bmatrix} \quad (8)$$

Notice that this seems like a conventional graph homomorphism except for the partiality of ε and υ, and for (7). This partiality is what allows us to define abstraction only on a portion of the original graph, that is, on the portion which is relevant for a particular application. This can be useful, for instance, in those cases in which certain regions are not significant for the purposes of a particular ATIS' user, i.e., the fleet management center of a deliver company that does not provide service to certain areas of a city, or a driver who never visits certain areas.

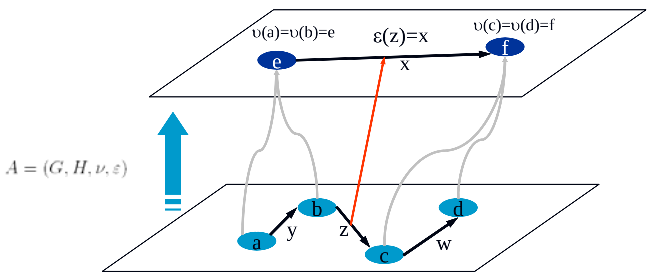

Figure 3. An example of topological abstraction of graphs. Abstraction of nodes and edges from one hierarchical level into another are defined by the functions υ and ε. Notice that due to the restriction imposed in (7), edges *y* and *w* can not be abstracted.

The node $\upsilon(a)$ for a given node $a \in V^{(G)}$ is the *supernode* of *a*, or the *abstraction* of *a*. Analogously, the edge $\varepsilon(z)$ for a given edge $z \in E^{(G)}$ is the *superedge* of *z*, or the abstraction of *z*. In the case that functions υ and ε of a given abstraction $A=(G,H,\upsilon,\varepsilon)$ are both total (that is, every node and every edge from graph G has a supernode and superedge, respectively), we will call *A complete*. In the case that both υ and ε are on-to functions (that is, nodes and edges from H are supernodes of superedge of any element of G), the abstraction A will be called *covered*.

Functions υ and ε have inverses, υ^{-1} and ε^{-1}, that yield the set of subnodes and subedges of a given node and edge respectively. They can be defined as:

$$\upsilon^{-1} : V^{(H)} \rightarrow 2^{V^{(G)}}$$
$$\forall b \in V^{(H)}, \upsilon^{-1}(b) = \{a \in V^{(G)} : def(\upsilon(a)) \wedge \upsilon(a) = b\} \qquad (9)$$

$$\varepsilon^{-1} : E^{(H)} \rightarrow 2^{E^{(G)}}$$
$$\forall y \in E^{(H)}, \varepsilon^{-1}(y) = \{z \in E^{(G)} : def(\varepsilon(z)) \wedge \varepsilon(z) = y\} \qquad (10)$$

For example, following the abstraction shown in figure 3, $\upsilon^{-1}(e) = \{a, b\}$, $\upsilon^{-1}(f) = \{c, d\}$, $\varepsilon^{-1}(x) = \{z, p\}$, and $\varepsilon^{-1}(r) = \{q\}$.

By constraints (6) and (7), an edge can not be abstracted if its incident nodes are not, or if its nodes have been abstracted to the same supernode (in that case, the edge "disappears" in the resulting graph *H*, as occurs with edges *y* and *w* in figure 3). Constraint (8) is the typical definition for graph homomorphism: when an edge is abstracted, the incident nodes of its superedge are the supernodes of the incident nodes of the edge (that is, connectivity is preserved). Sometimes we will need to refer to a component of an abstraction specifying explicitly the abstraction to which it belongs. For that purpose, the following notation is used in this paper: $G^{(A)}$, $H^{(A)}$, $\upsilon^{(A)}$, $\varepsilon^{(A)}$.

Notice that under particular conditions that can be found in most of the practical realization of this formulation (complete and covered abstractions), it is possible to determine the abstraction/refinement of edges by computing the abstraction/refinement of their incident nodes. Therefore, in the rest of the paper we only deal with abstraction/refinement of nodes.

4 Interactive Routing Process

In this section, the proposed interactive routing mechanism through multiple hierarchies is described. In this paper we assume a two-hierarchical representation of an urban map: one hierarchy, named the *routing hierarchy*, properly arranges topological information in order to search for routes efficiently, while the other one, the *user hierarchy*, is devoted to serve as a good interface with the user by considering her/his own labels and arrangements of topological information. The former can be automatically constructed (as in Galindo et al. 2007), while the latter must be obviously generated by the driver to capture her/his particular view and understanding of the urban map.

In our interactive routing process, the user inquires a route to a certain destination in terms of the information held by her/his hierarchy, i.e., "I want to go to the *Airport*". S/he expects to obtain the results also involving human-like concepts, albeit routing is efficiently performed through the *routing hierarchy*[5] by any hierarchical route planner/searcher (Jagadeesh et al. 2002, Quek and Srikanthan 2001, Huan and Jing 1996, Car and Frank 1993).

Typically, the *routing hierarchy* arranges spatial information in groups with no sense for the driver but in a way that makes route search efficient. Therefore, in order to achieve the proposed human interaction with the routing system some mechanism to translate information between hierarchies is needed. Next, a general process for translating information between hierarchies, called *inter-hierarchy translation*, is described. This mechanism is then applied to the particular case of the commented structure with two hierarchies (routing and user ones) to enable the driver to take decisions according to abstract information at different levels of detail, as detailed in section 4.2. Some implementation details are given in section 4.3.

4.1 The Inter-Hierarchy Translation Process

The *Inter-Hierarchy translation process* is aimed to translate topological information between any pair of hierarchies of a multi-hierarchical representation of urban maps. Broadly speaking, the translation mechanism consists of *refining* a certain node from a hierarchy down to the common ground level of the representation[6] (by means of the υ^{-1} function described in section 3.3), and then, *abstracting* the resultant ground nodes along the destination hierarchy through the function υ. While the abstraction of a given node through the function υ is unique, its refinement (υ^{-1}) is not, and therefore the

[5] The construction and the efficient use of the routing hierarchy are not considered in this work. Please refer to (Fernandez-Madrigal and Gonzalez, 2001).

[6] We assume that all hierarchies of the Multi-AH-graph share a common ground level: the most detailed representation of the city.

translation process may not produce a unique node at the target hierarchy, as commented further on.

To illustrate this inter-hierarchy translation process, let us consider a simple example in which two hierarchies represent topological information (for instance blocks of a city) under different user's criteria (see figure 4). This example does not include any hierarchy constructed explicitly for routing in order to show the generality of the inter-hierarchy translation process. In section 4.2 we will focus on the use of this process with specific routing hierarchies.

In the simple example of a multi-hierarchical representation of a urban map depicted in figure 4, one hierarchy, the *Zip Codes Hierarchy* (H_1), groups city blocks following the administrative division for postal service, and the other one, the *Police Station Hierarchy* (H_2), groups blocks covered by a given police station. In this example, an abstract route in terms of zip codes given to a policeman may not be useful enough and s/he could have a clearer idea of the areas to be traversed if the route is given in terms of other type of information, i.e., police stations. The opposite case would occur for a postman who would prefer the route given in terms of zip codes.

The formalization of the considered hierarchies for this example (see figure 4) is as follows (for simplicity we have not considered edges). Let the graphs (hierarchical levels) be:

$$V^{(G_0)} = \{b1, b2, b3, b4, b5, b6\};\ V^{(G_1)} = \{Z1, Z2\}$$

$$V^{(G_2)} = \{PS1, PS2, PS3, PS4\};\ V^{(G_3)} = \{SouthPS, NorthPS\}$$

where G_0 represents the blocks (*b1* to *b6*) of a portion of a city map, G_1 entails two topological concepts (*Z1* and *Z2*) for the two zip-code areas involved in the example, G_2, represents the four police stations (PS1 to PS4) that cover the considered area, and finally, G_3 groups police stations according to their location. Let the abstractions on these graphs be defined as:

$$A_0 = (G_0, G_1, \upsilon_0, \phi);\ A_1 = (G_0, G_2, \upsilon_1, \phi);\ A_2 = (G_2, G_3, \upsilon_2, \phi)$$

where

$$\upsilon_0(b1) = Z1; \upsilon_0(b2) = Z1; \upsilon_0(b3) = Z1; \upsilon_0(b4) = Z1; \upsilon_0(b5) = Z2; \upsilon_0(b6) = Z2$$

$$\upsilon_1(b1) = PS1; \upsilon_1(b2) = PS1; \upsilon_1(b3) = PS2; \upsilon_1(b4) = PS3; \upsilon_1(b5) = PS4; \upsilon_1(b6) = PS4$$

$$\upsilon_2(PS1) = SouthPS; \upsilon_2(PS2) = SouthPS; \upsilon_2(PS3) = SouthPS; \upsilon_2(PS4) = NorthPS$$

Hierarchy H_1 is a two-level hierarchy defined by abstraction A_0, while hierarchy H_2 is a three-level hierarchy defined by the abstraction chain $A_1 \bullet A_2$.

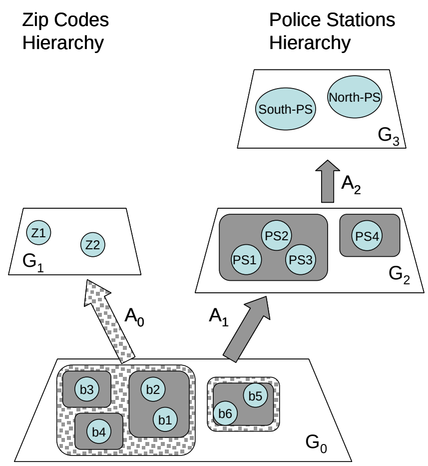

Figure 4. Example of a simple urban model with two hierarchies that group nodes from the ground level (i.e., blocks) with respect to its zip code (left) and the nearest police station (right). Filled regions remark groups of nodes abstracted to a common supernode: dotted regions for the zip codes hierarchy and solid areas for the police stations one. For clarity sake, edges have been obviated in this example, since they do not participate explicitly in the translation process.

In this example the translation of a certain node, let say *Z1*, from the *Zip Codes Hierarchy* (H_1) into the other hierarchy will provide the user with information about the police station(s) that cover all blocks that share the *Z1* zip code. Thus, for instance, if we consider a route involving nodes of H_1, (that is, information about the zip areas to be traversed) it could be translated in terms of the police station coverage along the route.

The information translation starts with the refinement of the considered node, i.e. *Z1*, yielding the set of its subnodes at the ground level (blocks in this case) belonging to the same administrative area, i.e., $[\upsilon_0]^{-1}(Z1) = \{b1, b2, b3, b4\}$.

Since the ground level is common to both hierarchies, subnodes of *Z1* can be abstracted now through the target hierarchy, yielding the set of police stations that offer service to each block within zip code *Z1*

$$\upsilon_1(b1) = PS1; \upsilon_1(b2) = PS1; \upsilon_1(b3) = PS2; \upsilon_1(b4) = PS3$$

Thus, the translation of the symbol *Z1* from the *Zip Codes Hierarchy* into the first level of the *Police Station Hierarchy* is the set $\{\upsilon_1(b1) \cup \upsilon_1(b2) \cup \upsilon_1(b3) \cup (b4)\}$, that is *{PS1,PS2,PS3}*, which can be abstracted again to *SouthPS*.

Observe that in this example the translation of *Z1* produces three nodes at the first level of the target hierarchy, but only one at the second level. This process provides the user with the knowledge that blocks with the zip code *Z1* are covered by *PS1, PS2,* or *PS3* police stations, and also that they all depend on the South Central Station (*SouthPS*).

4.2 Interactive Routing

Our interactive routing approach enables the driver to inquire a route to a destination specified in a particular and customized way through the topological information stored in her/his *user hierarchy* (for example, a user can command the system to go to "the nasty building", referring to her/his workplace). The route is firstly efficiently calculated by means of a given hierarchical routing algorithm (Fernandez and Gonzalez 2001) that exploits the information arrangement of the *routing hierarchy*. Information involved within the resultant route is then translated into concepts of the user hierarchy providing her/him with different interaction possibilities, as shown further on.

Let us consider the following example that illustrates our interactive routing process. We use a reduced multi-hierarchical representation of a city with two hierarchies, one automatically constructed for improving routing, for instance through one of the methods mentioned in (Galindo 2006, Fernandez and Gonzalez 2002, Knoblock 1994), and the other hand-maded by the driver involving her/his particular labels and groupings (see figure 5). For clarity sake, nodes from the ground level of this multi-hierarchy represent blocks, and therefore the resultant routes will involve partial paths between blocks without making explicit the street/avenue considered for such a path. That is, a route is understood here as a sequence of topological concepts. In a more extensive example, elements of this ground level would be the supernodes and superedges of more specific places as streets, junctions, buildings, etc. In such a case, the resultant route will entail a low-level route (in terms of streets and junctions) reported to the user through the correspondent labels stored in the driver hierarchy.

In this scenario, a driver wants to go to the *harbour* (B12) from his current location, his office (B1) at the *Business quarter* (see figure 5). A route to that destination is firstly calculated through hierarchical routing by successively abstracting the origin and destination upwards (the routing hierarchy) until reaching the highest level in which their supernodes are different. At this abstract level, in which probably a large amount of information has been abstracted, the original routing problem is efficiently solved by searching a route between the supernode of the origin to the supernode of the destination. This abstract route serves to ignore unnecessary topological information at lower levels when computing more refined solutions. It is demonstrated in (Fernandez and Gonzalez 2001), that for large graphs, hierarchical routing performs more efficiently than conventional flat routing.

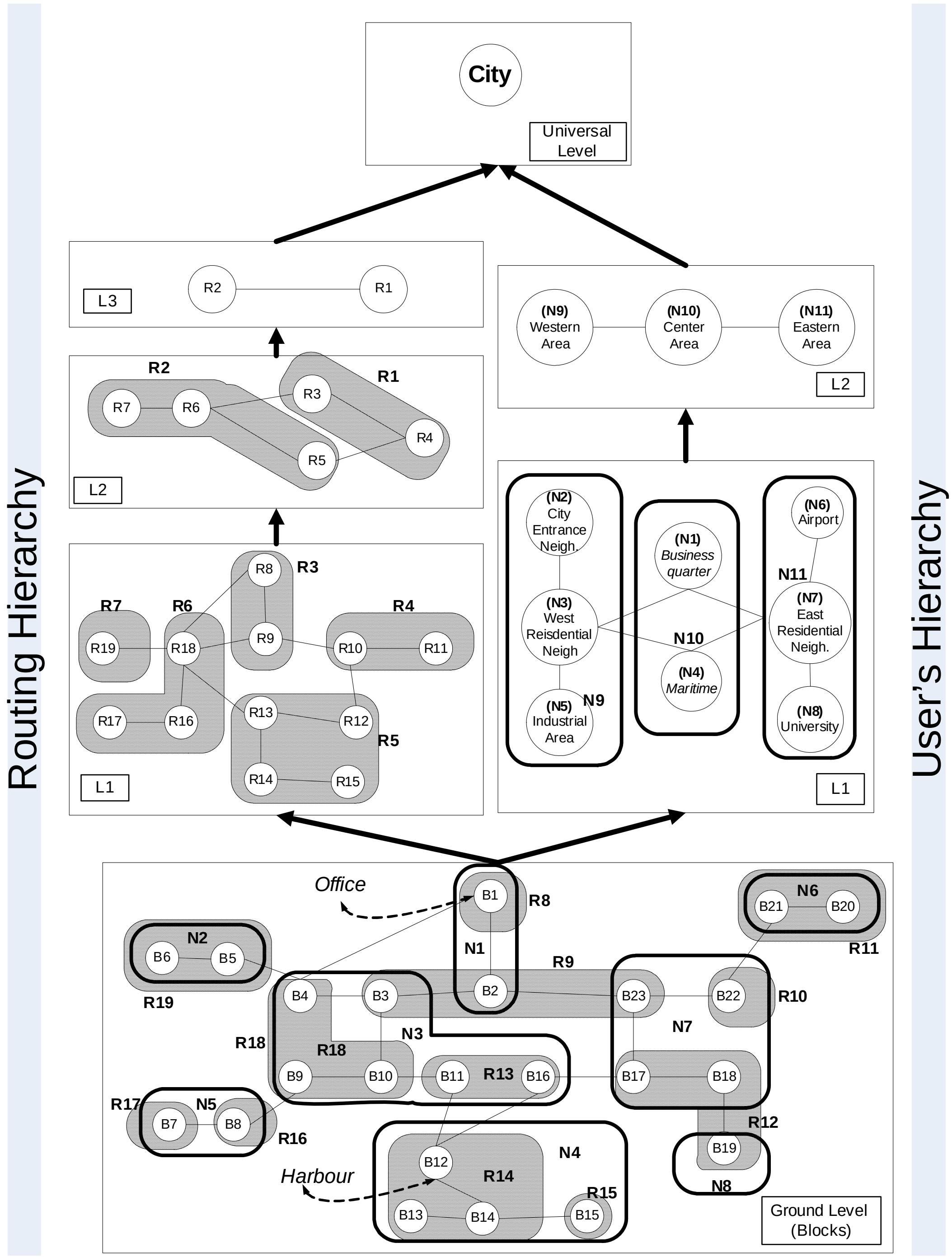

Figure 5. A Multi-AH-graph example for interactive urban routing. Upon a common ground level that represents base information (in this example blocks, noted as 'Bi' and their connections), two hierarchies are constructed for different aims: for efficiently routing (left) and for proper user-machine interaction (right). Regions indicate the different groupings at each level. For the routing hierarchy, textured regions with a short label in the form 'Ri' are used, while for the user's hierarchy we use non-textured regions with a short label that represents a name 'Ni'. These short labels are used for referring to groups of lower hierarchical levels.

In our example, the hierarchical routing process starts at level *L3*, being the supernode of the origin at this level *R1*, (calculated as $\upsilon(\upsilon(\upsilon(B1))$ and of the destination *R2*, and the abstract solution is "go from R1 to R2"[7].

Our interactive planning approach considers the communication of each intermediate solution of the hierarchical routing to the user in order to keep her/him fully informed about the details of the routing process. Thus, the first partial solution should be reported to the user. Since the particular arrangement of the information of the routing hierarchy is aimed to improve efficiency for routing (and not for human communication), concepts involved in routes, e.g. *R1*, may not have a logical meaning for humans, and thus they have to be translated into the *user hierarchy*.

As commented, the inter-hierarchy translation is achieved by refining successively route information. In this case, nodes from the *routing hierarchy* have to be translated into the ground level and then abstracted up through the *user hierarchy*.

The refinement of the abstract solution {(GO R1, R2)} down to the level L2 of the routing hierarchy involves all the possible combinations of the subnodes of its parameters: *R1* and *R2*, that is $\left\{\left(GO\,\upsilon^{-1}\left(R1\right),\ \upsilon^{-1}\left(R2\right)\right)\right\}$, which can be expressed as {(GO {R3,R4}, {R5,R6,R7})}, as long as $\upsilon^{-1}\left(R1\right)=\left\{R3,R4\right\}$, and $\upsilon^{-1}\left(R2\right)=\left\{R5,R6,R7\right\}$. This refinement implies that the route can be calculated at the second level of the routing hierarchy by only considering R3 or R4 as the origin and R5,R6, or R7, as the destination.

Refining this to L1, we obtain:
$\left\{\left(GO\,\upsilon^{-1}\left(R3\right)\cup\upsilon^{-1}\left(R4\right),\ \upsilon^{-1}\left(R5\right)\cup\upsilon^{-1}\left(R6\right)\cup\upsilon^{-1}\left(R7\right)\right)\right\}$, that is:
{(GO {R8,R9,R10,R11}, {R12,R13,R14,R15,R16,R17,R18,R19})}, and finally, its refinement at the ground level yields: {(GO {B1,B2,B3,B23,B22,B20,B21},
{B17,B18,B19,B11,B16,B12,B13,B14,B15,B8,B7,B4,B9,B10,B5,B6}

(11)

[7] In the following, the command, like (GO R1,R2), is used to refer to the path "from R1 to R2" returned by the routing system.

This solution, which does not offer any information to the user, implies to go from one part of the city (that includes the origin *B1*) to another part of the city (that includes the destination *B12*). Since the abstract solution computed at level L3 has been refined until the common ground level, it can be abstracted now through the user hierarchy enabling the user to be informed about how the routing process is being carried out.

The abstraction of the route from the ground level to the first level of the user hierarchy is computed by applying the function υ to its parameters:

$\{(\text{GO } \{\upsilon(\text{B1})\cup\upsilon(\text{B2}),...\upsilon(\text{B20})\cup\upsilon(\text{B21})\},\{\upsilon(\text{B17})\cup\upsilon(18),...\cup\upsilon(\text{B5})\cup\upsilon(6)\})\}$ which can be expressed as[8]:

```
(GO Bussiness-Quarter, {City Entrance, West Residential, Maritime,
Industrial, East Residential, or University})
```

(12)

and at level 2, as:

```
(GO Center, {Western, Center, or Eastern Areas})
```

Notice the vagueness of the translated route communicated to the user at the second level of the user hierarchy, and the ambiguity of the route at level 1. This is because a very early solution computed at a high abstracted level of the *routing hierarchy* has been translated and communicated to the user. In this situation she/he can proceed in the following ways:

- *Inquire a more detailed route*. The driver can reduce the information ambiguity by inquiring a route computed at a lower level of the *routing hierarchy*. Thus, the routing algorithm constructs a more specific route (involving more detailed information but also with a higher computational burden). For instance, the route calculated at the lower level of the *routing hierarchy* is {(GO R3, R6) (GO R6, R5)}, and at the first level {(GO R8, R18) (GO R18 R13) (GO R13, R14)}. This last route can be translated at the highest level of the user hierarchy as: {(GO *Center-Area, Western-Area*) (Go *Western-Area, Western-Area*) (GO *Western-Area*, *Center-Area*)}, being the route communicated to the user: {(GO *Center-Area*, *Western-Area*) (GO *Western-Area*, *Center-Area*)} in which partial routes

[8] Note that since the origin is given in terms of the user hierarchy, it does not require any translation.

with the same origin and destination are discarded. In this case the driver only knows that the *Eastern-Area* of the city is not considered for going to the *harbour*. For a more detailed route, s/he can inquire further translations at lower levels of the user hierarchy. Thus, the translation of the computed route {(GO R8, R18) (GO R18 R13) (GO R13, R14)} at the first level of the user hierarchy is: {(GO *Bussiness-Quarter, West-Residential*) (Go *West-Residential, Maritime*)}.

- *Reject part of a route*. Observe that even when the provided route does not reveal enough information, the user can interact productively with the routing process, i.e. by rejecting certain parts of it. For example when the driver is provided with the translation of the abstract route (12), s/he may want to avoid the neighbourhoods *West-Residential* and *Industrial* for some reason. The discarded nodes are translated into the *routing hierarchy* by means of inter-hierarchy translation, indicating to the routing process that the nodes *B3, B4, B7, B8, B9*, *B10*, and *B11* at the ground level, *R16*, *R17*, and *R18* at level 1, and *R6* at level 2 must not be longer considered. Then a new route is calculated at that level, producing the new route {(GO R3, R4), (GO R4, R5)}, which is reported to the driver as {(GO *Business-Quarter*, {*East-Residential or University*}), (GO {*East-Residential or University*}, {*East-Residential or Harbour*})}.

- *Suggest a route*. As commented, the user may be informed about a set of different possibilities to arrive at a destination at any level of the *user hierarchy*. In these cases the user could decide to select one out of the offered routes based on her/his knowledge of the environment or feelings. For instance, the driver can suggest the route: {(GO *Business-Quarter, East-Residential)* (GO *East-Residential, Harbour*)} since she/he does not wish to consider the crowded *University* neighbourhood. Thus, through the solution pointed out by the driver, the routing algorithm can calculate a path at the ground level considering only the information embraced by the areas suggested by her/him and thus reducing the computational cost. In this example, the final route at the ground level is

{(GO *Office* B2), (GO B2 B23), (GO B23 B17), (GO B17 B16), (GO B16 *Harbour*)}[9].

Notice that, apart from the benefits for user interaction, user participation into the routing process may also improve efficiency of the hierarchical routing process. In hierarchical routing, efficiency is normally achieved by discarding unnecessary information with respect to the problem at hand. Now, an additional mechanism to discard useless information (from the point of view of the driver) can be provided by user interaction: when the user rejects or accepts a route (or part of it), she/he may discard topological information, probably implying further simplification of the routing process.

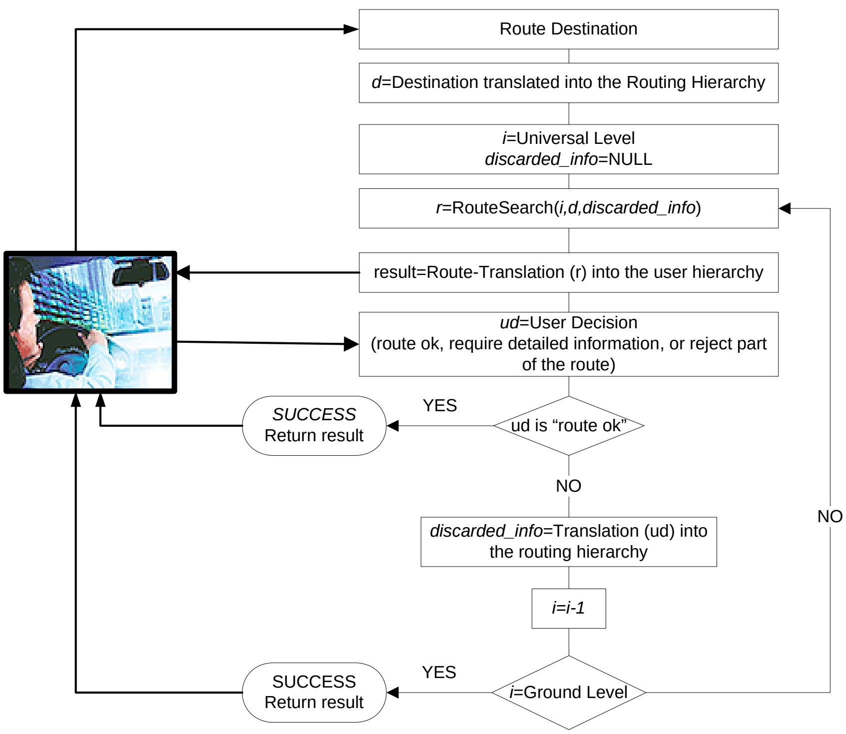

Figure 7. Interactive Routing Scheme. Routes to a destination are given to the user at different levels of detail, enabling her/him to actively interact with the process.

[9] A part from the origin and destination, intermediate nodes should also have a particular label understandable by the driver, indicated here as *Bxx* for clarity.

In this process, and depending on the structure of the hierarchies, abstract routes might involve a large number of possible locations, like in (11), hampering the communication to the driver. In our approach routes entailing more than five optional places are not reported to the user, but a more refined route at the routing hierarchy is automatically computed and communicated to the driver.

4.3 Implantation details

Figure 7 sketches the proposed interactive routing process which has been tested through a graphical interface (see section 5). However, the development of an in-vehicle system featured with our approach would require a more sophisticated human-system interface, i.e. a voice system enabling user interaction through verbal utterances. Such an interaction within a noisy and unpredictable environment may become complex and is out of the scope of this work.

We want to remark that our interactive routing approach can also fit in a more general scheme in which a commercial GPS-based guiding tool substitutes the routing hierarchy, providing routes in terms of low-level information (streets), i.e. at the ground level of the *user hierarchy*. In this case, the translation process would only consist of abstracting ground information through the *user hierarchy* to report abstract routes to the driver, as well as refining abstract information to indicate the streets to be avoided. Notice that in that case, the multihierarchy can still be useful, for example if more than one driver use the same device. Also, it is important to highlight that although we have focused on the use of our approach in in-vehicle applications, this work can also be used in other scenarios, for example, people interacting with a GIS and expecting human-understandable results.

Next section illustrates our interactive routing approach within a real scenario, concretely the city of Málaga, in Spain.

5 A case of Study (MÁLAGA)

The Multi-AH-graph model and our interactive routing approach has been implemented in C++ and tested through a portion of the topological information of Málaga (Spain) using ArcGIS. User interaction and communication has been implemented through graphical/textual interfaces. We have used a ground level containing around 10,000 nodes representing junctions and buildings entrances. Upon this ground level the foresaid two hierarchies have been created: the *user hierarchy* is manually constructed to group topological information into human understandable concepts under the particular point of view of a typical driver, and the *routing hierarchy* is automatically constructed for yielding efficient routes connecting the most visited places by the user. This automatic construction is not addressed here and the reader can consult (Fernandez and Gonzalez, 2002) for more explanation.

The scenario we consider for this case of study is Málaga, a growth city located at the south of Spain with over half a million of citizens. It becomes the administrative center of a large region which hinges on the airport and harbour of Málaga for its economical development. Traffic congestion is a daily problem due to the quick rise in the number of automobiles (it has been tripled in the last decade) and the continuous roadwork. Let us describe the case of study in which a driver goes to Málaga for the first time to catch a pleasure cruise. He has no knowledge about the city except from two sources: the information he had read in a tourist guide and the advices given by a friend. Using the guide, the user can construct a user hierarchy which, in fact, would serve for drivers in general and thus, it is assumed to be pre-loaded in the system. This hierarchy presents the city under standard topological information. Moreover, the user can modify it following the advices given by his friend who utilized the same topological labels found in the guide:

- *Miraflores* is the best place to enter the city with many good restaurants.
- The *Trinity* area of Málaga is enchanting, plenty of historical monuments, but becomes a traffic chaos at rush hours. It is the same for the East part of *Carranque*.

- The *Cruz-Verde* area has excellent roads, although it has a high criminality rate; I recommend you to avoid it, if possible, as well as the North part of the *Old Town*.
- The train station is at the *Cuarteles* area, where a dirty industrial area is also located.

Accordingly to these advices, the driver modifies the user hierarchy provided by the guide. Notice that in this case it can be considered that the driver borrows the user hierarchy of his friend with his particular feelings and experiences. As long as the user comes into contact with the city he can include his own modifications. In our experiences, the process for creating the *user hierarchy* (or modifying an existing one) is carried out by the driver through a graphical tool on ArcGis that allows him/her to select a set of nodes and to group them into a supernode at the higher hierarchical level (see figure 8).

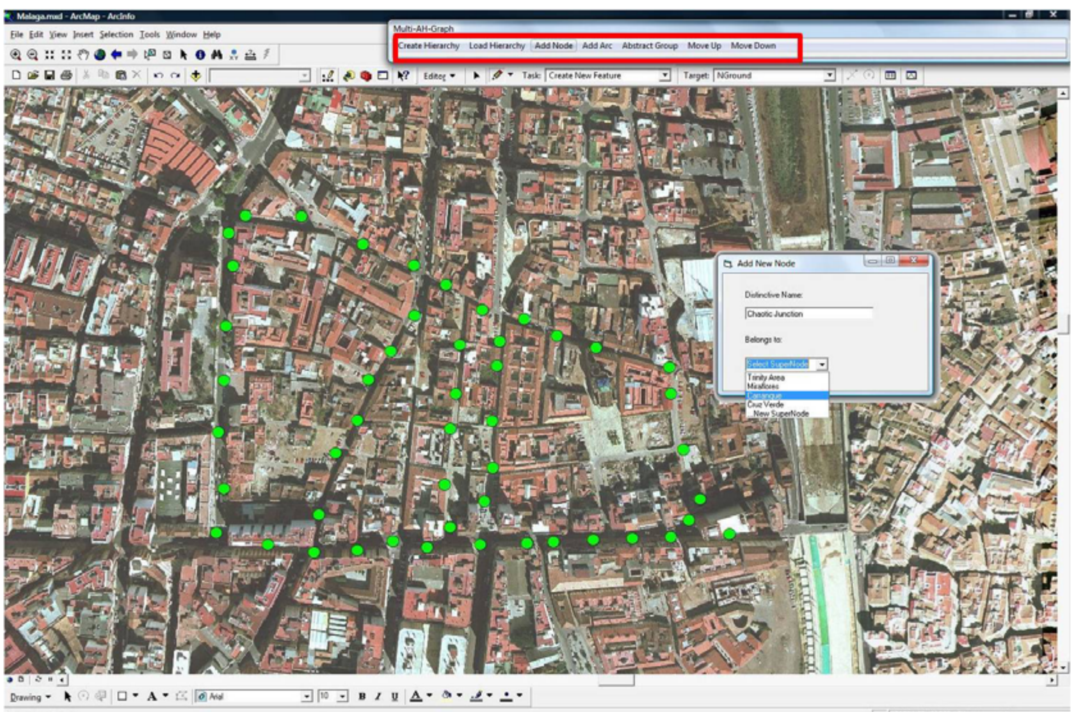

Figure 8. Creation of a user hierarchy on Arcgis. The implemented plug-in permits the user to create new nodes (through the dialog shown in the figure), as well as a number of utilities such as grouping a set of nodes into the same supernode, moving up and down through the hierarchy, etc.

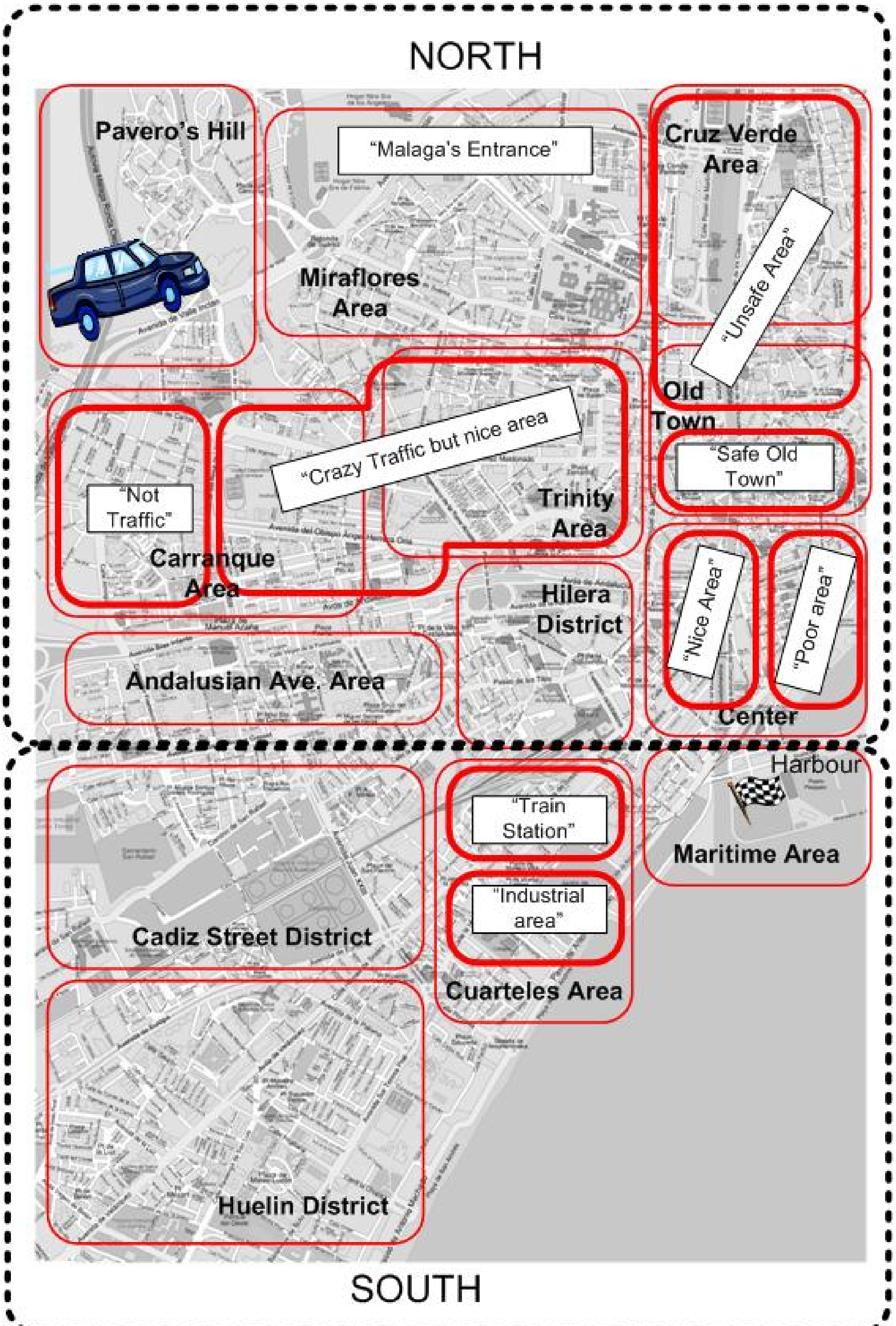

Figure 9. Map of Málaga (Spain). Administrative areas appear in bold and their limits with thin lines, corresponding to the grouping provided by the tourist guide. Modifications made by the driver are shown with thick lines with their respective labels. Dark thin route indicates the fastest way to achieve the harbour. Dark thick line shows the route calculated by the interactive routing system to fulfil the driver requirements.

Considering the user modifications made to the user hierarchy provided by the guide (see figure 9), the s/he asks the system to go to the *harbour*, which is at the *Maritime Area*, i.e., the harbour is a subnode of the node that represents the Maritime Area in the *user hierarchy*.

Through the *routing hierarchy* not shown here for simplicity, the considered hierarchical path finder provides the user with a sequence of routes: firstly, a non-relevant route is given: *(GO to the South)*. Such information does not endow the driver with any cue about the path to follow, so he inquires a more detailed route. Then, the system considers more refined information (at lower levels of the *routing hierarchy*) yielding the following set of possible routes (ordered with respect to their expected travel time; the higher it appears, the quicker it is):

Route 1:	"Malaga's Entrance" → "Unsafe Area" → "Safe Old Town" →{"Nice zone" or "Poor and dirty buildings area"} → Maritime
Route 2:	"Malaga's Entrance", → "Unsafe Area" → "Safe Old Town" → "Crazy traffic but nice area" → Hilera → Maritime
Route 3:	"Malaga's Entrance" → "Unsafe Area" → "Safe Old Town" → "Crazy traffic but nice area" →Hilera → {"Train Station" or "Dirty Area"} → Maritime

Notice that at this level all routes involve the district *Cruz-Verde* (called by the user "Unsafe Area") since it is the quickest way to arrive at the destination. Following the advices of his friend, the driver asks the system for alternative paths not involving that area, and therefore, any of its streets. The set of routes, once they are translated to the user hierarchy, are now:

Route 4:	{"Crazy traffic but nice area" or "Not traffic at all"} → Andalusian Ave. → Hilera → Maritime
Route 5:	"Malaga's Entrance" → "Crazy traffic but nice area" → Hilera → Maritime
Route 6:	"Malaga's Entrance" → "Crazy traffic but nice area" → Hilera → {"Train Station" or "Dirty Area"} → Maritime

Route 4 seems to be the quickest and safety route to go to the harbour, albeit the driver could not be completely agree with it if he prefers to go to one of the restaurants located at the entrance of Malaga and to visit the nice area of *Trinity*, as his friend recommended. Thus, he informs the interactive system that the desired path is *Route 5*,

in spite of not being the quickest one (route 6 is discarded since it involves the possibility of passing through the “dirty area”). Once a particular route has been suggested by the driver, the final path at the ground level of the multi-hierarchy, involving streets and junctions, is generated to guide the driver to the destination. Both the quickest route and the one selected by the user are depicted in figure 9 as well as the user hierarchy provided by the guide and the modifications made by the driver.

6 Conclusions and Future Work

This paper has presented an interactive in-vehicle routing system that permits the driver to choose a route to a given destination that fulfils her/his personal requirements. Routes are reported to the driver by using high-level and understandable information from which s/he can productively interact with the system. At each step the driver can interact with the system considering the provided high-level information, for instance name of districts, to take a prompt decision: the driver can accept an abstract route, since s/he does not need further information and thus finishing the routing process, or s/he can reject part of an abstract route due to a variety of causes which may not be captured by conventional routing algorithms. In this latter case, the proposed interactive system searches for alternative routes that fulfil the driver requirements.

Our interactive routing approach has been designed and implemented through a multi-hierarchical representation of spatial data. Concretely, an illustrative example of our technique has been presented in the case of driving through Málaga city.

In the future we plan to integrate commercial GPS-based routing tools with our approach in real experiences in order to test the driver satisfaction when using our system.

References

Azevedo J., Costa M.S., Madeira J.S., and Martins E.V. An algorithm for the ranking of shortest paths. European Journal of Operational Research. Vol. 69, pp. 97-106, 1993.

Ben-Akiva M., Bergman M., Daly A., and Ramaswamy R. Modeling inter-urban route choice behaviour. Proc. Of the 9th Int. Symposium on Transportation and Traffic Theory, VNU Science Press, Utrecht, Netherlands, pp. 299-330, 1984.

Bierlaire M., Frejinger E. Route Choice Models with Subpath Components. Conference paper Swiss Transport Research Conference, Ascona, March 9-11, 2005.

Bolduc D. and Ben-Akiva M. A multinomial probit formulation for large choice sets. Proc. Of the 6th Int. Conf. on Travel Behaviour, Vol.2, pp. 243-258, 1991.

Car A. and Frank A.U.. Hierarchical street networks as a conceptual model for efficient way finding. Proc. Fourth European Conf. and Exibiton on GIS, vol. 1, 1993, pp. 134-139.

Cascetta E. and Papola A. Random utility models with implicit availability/perception of choice alternatives for the simulation of travel demand, Transportation Research Part C 9: 249–263, 2001.

Dingus T.A., Hulse H.C. Some human factors design issues and recommendations for automobile navigation information systems. Transportation Research 1C(2), 119-132, 1993.

Fernández-Madrigal J.A. and González J., Multi-Hierarchical Representation of Large-Scale Space. International Series on Microprocessor-based and Intelligent Systems Engineering. Vol. 24, Kluwer Academic Publishers, Netherlands, 2001.

Fernández-Madrigal J.A. and González J., Multihierarchical graph search, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24 (2002), pp. 103–113.

Fernández-Madrigal J.A., Galindo C., and González J., Assistive navigation of a robotic wheelchair using a multihierarchical model of the environment, Integrated Computer-Aided Engineering, 11 (2004), pp. 309–322.

Fernández-Madrigal J.A.. A Mathematical Formulation of Not-Restricted, Multiple Abstraction Based on Graphs. Application to Graph Isomorphism. Tech. Report, System Engineering and Automation Dpt., University of Málaga, 2004.

Galindo C., Fernandez-Madrigal J.A., and Gonzalez J., Hierarchical task planning through world abstraction, IEEE Trans. on Robotics, 20 (2004), pp. 667–690.

Galindo C., González J., and Fernández-Madrigal J.A. A Control Architecture for Human-Robot Integration: Application to a Robotic Wheelchair. IEEE Transactions on Systems, Man, and Cybernetics Part B, vol. 36, no. 6, pp. 1053–1068, 2006.

Galindo C. A Multi-Hierarchical Symbolic Model of the Environment for Improving Mobile Robot Operation. PhD Thesis, System Engineering and Automation Department, University of Malaga (Spain), 2006.

Galindo C., Fernandez-Madrigal J.A., Gonzalez J., Saffiotti A., and Buschka P. Life-Long Optimization of the Symbolic Model of Indoor Environments for a Mobile Robot. IEEE Trans. On Systmes, Man and Cybernetcis, Part B. vol. 37, no.5, pp. 1290-1304, 2007.

Holding C.S. Further Evidence for the Hierarchical Representation of Spatial Information Journal of Environmental Psychology, Vol 14., 1994.

Huang Y. and Jing N. Evaluation of hierarchical path finding techniques for ITS route guidance. Proc. Annual Meeting of ITS America, vol. 1, 1996, pp. 340-350.

Huang Y-W, Jing N., and Rundensteiner E.. A Hierarchical Path View Model for Path Finding in Intelligent Transportation Systems. Geoinformatica, vol.1 no. 2, pp. 125-159, 1997.

ISO TC204 WG1, Transport Information and Control Systems - Reference Model Architecture(s) for the TICS Sector - Part 1: TICS Fundamental Services, ISO/TR 14813-1, 1998.

Jagadeesh G.R., Srikanthan T., and Quek K.H.. Heuristic Techniques for Accelerating Hierarchical Routing on Road Netwoks. IEEE Trans. On Intelligent Transportation Systems, vol. 3, no. 4, pp.301-309, 2002.

Jagadeesh G.R., Srikanthan T., and Quek K.H. Combining hierarchical and heuristic techniques for high-speed route computation on road networks. Computing & Contro Engineering Journal, June 2002.

Jones C.B. and Zhou S. Design and Implementation of Multi-scale Databases. In 7th Int. Symposium on Spatial and Temporal Databases, California, pp. 365-386, 2001.

Jong-Hyun P., Wikantika K., and Tateishi R.. Comparison of four different methods to fuse multi-sensor and multi-resolution remotely sensed data for urban topographic mapping. Geoscience and Remote Sensing Symposium, 1999. Vol.2, pp.1203 – 1205.

Khattak A.J., Shofer J.L., and Koppelman F.S. Commuters' enroute diversion and return decisions: analysis and implications for advanced traveller information systems. Transportation Research 27A(2), 101-111, 1993.

Knoblock C.A. Automatically Generating Abstractions for Planning. Artificial Intelligence, Vol. 68, No. 2 pp. 243-302, 1994.

Mackaness W., Ruas A., Sarjakoski L. Generalisation of Geographic Information: Cartographic Modelling and Applications. International Cartographic Association Series, 2007.

Mannering F., Kim S.C., Barfield W. Traveler's preferences for in-vehicle information systems: an explanatory analysis. Transportation Research 3C(6), pp. 339-351, 1995.

McNamara T.P., Hardy J.K., and Hirtle S.C. Hierarchies in Spatial Memory Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol 15, 1989.

Miller J.H.and Shin-Lung S.. Geographic Information Systems for Transportation. Oxford University Press, 2001.

Nielsen O., Daly A., and Frederiksen R. A stochastic route choice model for car travellers in the Copenhagen region. Networks and Spatial Economics, vol. 2, pp. 327-346, 2002.

Paiva A.C., da Silva E.R., Leite F.L., and de Souza Baptista C.. A multiresolution approach for Internet GIS applications. Proc.. 15th Int. Workshop on Database and Expert Systems Applications, 2004, pp.809 – 813.

Patalano A.L., Smith E.E., Jonides J., and Koeppe R.A.. PET Evidence for Multiple Strategies of Categorization. Cognitive, Affective, and Behavioral Neuroscience, Vol.1, N. 4, 2001.

Quek K.H. and Srikanthan T., "Improving Hierarchical Route Computations for Roadway Networks", 4th International IEEE Conference on Intelligent Transportation Systems, pp.112-117, 2001.

Ross T., Vaughan G., and Nicolle C. Designe guidelines for route guidance systems: development process and an empirical example for timing of guidance instructions. Ergonomics and Safety of Intelligent Driver Interfaces. Lawrence Erlbaum Associates, pp. 139-152, 1997.

Scott K., Pabon-Jimenez G. and Bernstein D., Finding alternatives to the best path, Proc. 76th Annual Meeting of The Transportation Research Board, Washington, D.C. Paper No. 970682, Jan. 1997.

Smith B. and McGinty L. The Route to Personalization: A Case-based Reasoning Perspective. International Conference on Knowledge Based Systems and Applied Artificial Intelligence, 2001.

Thomas R. and Donikian S.. A Model of hierarchical cognitive map and human-like memory designed for reactive and planned navigation. Proc. 4th Int. Space Syntax Symposium, London 2003.

Timpf S. and Frank A.U. Using Hierarchical Spatial Data Structures for Hierarchical Spatial Reasoning. Lecture Notes in Computer Science 1329, pp. 69-83, 1997.

Timpf S., Volta G.S., Pollock D.W., and Egenhofer M.J.. A Conceptual Model of Wayfinding Using Multiple Levels of Abstractions. In: Frank, A.U.; Campari, I.; Formentini, U. (Eds.), Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Lecture Notes in Computer Science, 639. Springer Verlag, Heidelberg-Berlin, pp. 348-367, 1992.

Timpf S. and Kuhn W. Granularity Transformations in Wayfinding. In: C. Freksa, W. Brauer, C. Habel and K.F. Wender (Eds.), Spatial Cognition III. Lecture Notes in AI. Springer Verlag, Berlin-Heidelberg, pp. 77-88, 2003.

Wochinger K. and Boehm-Davis D. Navigational preference and driver acceptance of advanced traveller information systems. Ergonomics and Safety of Intelligent Driver Interfaces. Lawrence Erlbaum Associates, pp. 345-362, 1997.