# A Geospatial World Model for the Semantic Web

**A Position Paper**


François Bry[1], Bernhard Lorenz[1], Hans Jürgen Ohlbach[1] and Mike Rosner[2]

[1] Institute for Informatics, Ludwig-Maximilians University Munich,
{bry,lorenz,ohlbach}@pms.ifi.lmu.de
[2] Department of Computer Science and AI, University of Malta,
mike.rosner@um.edu.mt



**Abstract.** The Semantic Web is an endeavour aiming at enhancing Web data with meta-data and data processing, as well as processing methods specifying the "meaning" of such data and allowing Web-based systems to take advantage of "intelligent" reasoning capabilities. The representation of the meaning of data essentially requires the development of a world model. Ontologies, for example, are logical descriptions of world models. In this paper we investigate what it means to develop a world model for "geospatial" data that can be used for Semantic Web applications. Different aspects are analysed and a proposal for a concrete architecture is developed. The architecture takes into account that geospatial data (road maps etc.) are usually owned by companies and only accessible through their interfaces. The article also argues that, to complement standard, general purpose, logic-based data modelling and reasoning methods, as e.g. offered by RDF and OWL and reasoners for these languages, location reasoning is best tackled using graphs for data modelling and well-established algorithms for reasoning. Hence, the article illustrates, for the practical case of location reasoning for providing guidance, the thesis that, on the Semantic Web, "theory reasoning" is a desirable complement to "standard reasoning".


## 1 Introduction

The Semantic Web is an endeavour aiming at enhancing Web data with meta-data and data processing, as well as processing methods specifying the "meaning" of such data and allowing Web-based systems to take advantage of "intelligent" capabilities. In a Scientific American article [1] which has diffused the Semantic Web vision, this endeavour is described as follows:

> "The semantic web will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users."

Reasoning is central to the Semantic Web vision since reasoning is central to processing *declarative* data and specifying *intelligent* forms of data processing. In the abovementioned Scientific American article, this central role of reasoning for realizing the Semantic Web vision is stressed as follows:

> "For the semantic web to function, computers must have access to [. . . ] sets of inference rules that they can use to conduct automated reasoning." [1]

Inference rules operate on facts and axioms. Axioms specify in an abstract way a model of the world. For example, the axiom $\forall x\ motorway(x) \Rightarrow road(x)$ says something about the relation between the words 'motorway' and 'road'. The most detailed axiomatisations which are currently being used for the Semantic Web are ontologies. They are formulated in logical formalisms like Description Logics [2] or OWL [3] and describe more or less complex relationships between different notions (concepts and relations) used in particular domains. Pure logical formalisms have a somewhat one-track style of expressiveness, so logical axiomatisations often give only a very coarse picture of the world. A web service, for example, which computes the shortest way to get from Munich to Hamburg needs a much more detailed picture of the world, namely digital road maps, than any pure logical axiomatisation is likely to provide.

In this paper we argue that "geospatial" notions play an important role for the Semantic Web, and that a very sophisticated world model is necessary for giving them a useful semantics. The world model consists of concrete data, road maps, train connections, floor plans etc., as well as logically formalised ontologies of, for example, transport networks. We sketch a first approach which combines concrete computations with data from Geographical Information Systems (GIS), for example route planning, and higher level logical formalisations. Our approach also takes into account very practical constraints, such as companies owning and not releasing GIS data.

We also argue that to complement standard, general purpose, logic-based data modelling and reasoning methods, as e.g. offered by RDF and OWL and reasoners for these languages, geospatial reasoning with topographical data is best tackled using graphs for data modelling and well-established graph algorithms for handling inference.

Completely general reasoning techniques must, by their very nature, be weakly committed to any particular class of problems and are thus unable to take advantage of any particular properties of that class. We therefore claim not only that the class of geospatial reasoning problems requires equally specific reasoning methods but that logic-based, general-purpose methods could never properly, intuitively, and efficiently realize what is best achieved using graphs and graph algorithms.

It has been claimed by Bry and Marchiori [4] that, on the Semantic Web, "theory reasoning" is a desirable complement to "standard reasoning". This articles substantiates this claim with respect to evidence from the practical case of geospatial reasoning for geographical guidance.

## 2 Motivating Examples

Before we present our approach we illustrate potential applications with simple examples and case studies. The first group of examples concerns querying XML or ordinary databases.

*Example 1.* Suppose we have some data about cities, states and countries. Entries could be:

1. San Francisco is a city
2. San Francisco is in California
3. San Francisco has 3 million inhabitants
4. California is in the USA.

A query could be: "give me all metropolises in the USA". In order to evaluate this query we need to:

- formulate the database entries in a logic based knowledge representation language, for example OWL or its underlying Description Logic.
- define the concept "metropolis" in the same knowledge representation language, e.g.
$$metropolis = city \wedge atleast\ 1000000\ has\_inhabitant \tag{1}$$
(A a metropolis is a city with at least 1 million inhabitants.)
- make a so called *instance test* for the database entries. The instance test would conclude from (2) and (4) that San Francisco is in the USA, and from (1) and (3) that San Francisco is a metropolis. ■

*Example 2.* Suppose the database contains the yellow pages entries, i.e. businesses with their addresses. A query could be: "give me the nearest pharmacy", with the context information that I am at a particular location $X$ in the city, and with all the other context information about my current situation (availability of a car, luggage, my age and gender etc.).

This query could be evaluated in a naive way by selecting the pharmacy with the smallest geographic distances between it and the location $X$. This might be a first approximation, but it can give completely useless results. A pharmacy which is located very close by, but unfortunately it is on the other side of the river, and the next bridge is miles away, may not be a good choice.

The answers would be much more appropriate if we use, instead of the geographic distance, a metric which is determined by the local transport systems. This means, the nearest pharmacy is the one which can be reached in the shortest time. This problem amounts to a route planning problem. The system must compute the shortest route from the location $X$ to the pharmacies and choose the one with the shortest route. The route planner must take into account the transport networks (road maps, tram lines, bus lines etc.), as well as the context information about the users current situation.

Reasoning about locations normally operates at a numerical level (e.g. coordinates) or at a symbolic level (e.g. graphs). Extensive research has been conducted in either case [5], hence there is a broad choice of proven sets of calculi and algorithms to solve the respective tasks [6–9]. The fundamental insight is that many queries pertaining to location information are closely related to the problem of route planning and way finding. There are two reasons for this. First, whenever a certain location is sought after, the chances are that the inquirer intends to visit the location. Cases like these result in classic route planning tasks. Second, when people refer to the "distance" between two locations in the sense of locomotion, they are almost never talking about distances per se (metres, kilometres) but the time needed to cover these distances ("a ten minute walk" or "half an hour by train"). In fact, in many scenarios the absolute distance between two points is of rather marginal significance from a traveller's point of view, especially in urban environments.

As stated in section 1, general purpose reasoning is not the ideal choice for more complex reasoning tasks like route planning which involve a number of locations and/or additional constraints. Of course, general purpose reasoning can be used for some subtasks, such as deriving from the symbolic information shown in figure 5, that for example "Munich" is located in "Germany" (since it is located in "Bavaria", which in turn is part of "Germany"). More complex tasks, such as finding out which pharmacy or hospital can be reached in the shortest time involves a number of subtasks and higher level reasoning techniques. ■

*Example 3.* Consider the query "give me all cities *between* Munich and Frankfurt". What does *between* mean here? If we take a map of Germany and draw a straight line from Munich to Frankfurt, it does not cross many cities. A more elaborate (and still too simple) formalisation of *between* could be: in order to check whether a city $B$ is between the cities $A$ and $C$, compute the shortest route $R_1$ from $A$ to $B$, the shortest route $R_2$ from $B$ to $C$ and the shortest route $R_3$ directly from $A$ to $C$. If the extra distance $d = length(R_1) + length(R_2) - length(R_3)$, I need to travel from $A$ to $C$ via $B$, compared to the direct route from $A$ to $C$, is small enough, $B$ can be considered to be *between* $A$ and $B$. Since the condition "is small enough" is not very precise, one could use the distance $d$ directly to order the answers to the query. ■

*Example 4.* Suppose a company looks for a building site for a new factory. The site should be *close to* the motorway. "Close to" does in this case of course not mean the geographic distance to the motorway. It means the time it takes for a car or for a lorry to get to the next junction of the motorway. The length of the shortest path to the next junction can be used to order the answers to the query. ■

*Example 5.* Suppose the database contains a road map, together with dynamic information about, say, traffic jams. The information about traffic jams is usually not very precise. It could be something like "there is a traffic jam on the M25 2 miles long between junction 8 and junction 10".

If the M25 is taken as a straight line then the traffic jam is a one-dimensional interval whose location is not exactly determined. Instead, we have some constraints: length = 2 miles, start after coordinate of junction 8, and end before coordinate of junction 10.

So queries like "is there a traffic jam on the western part of the M25" give rise to a constraint-solving problem. ■

The ability to solve route planning problems is obviously very important for a useful geospatial world model. If this is solved, and there are good solutions already available, one can think of more interesting examples.

*Example 6 (Appointment Scheduling).* For a route planning algorithm it makes no difference if a route is to be planned such that a traveller catches, say, a particular train in a particular train station, or that he meets a particular person in his office. Appointment scheduling with a single person is therefore an instance of a route planning problem. More interesting are problems where several persons want to meet at a particular place.

In this case one has to solve two problems. The first problem is to find the time slots where they can meet. This is a constraint handling problem. The second problem is to synchronise the routes of the different persons such that they really meet at their meeting place. ■

## 3 Practical Constraints

A useful geospatial world model needs geographical data of various kinds, road maps, public transport networks, floor plans of buildings, where the books are in the bookshelves of libraries, or where the items are on the shelves of supermarkets etc. This data are owned by various companies and organisations: the government which operates the highways or the public transport systems within a city, the company that runs an airline or a taxi service, or the owner of a building. Some companies have built up large databases of geographical data and earn money by granting limited access to them. Companies like NAVTEQ [10] or Tele Atlas [11] operate and maintain databases about infrastructures, which other parties (governments, companies) are responsible to build, maintain and operate. NAVTEQ, for example, took some seven years to build their database about the German road and highway network, which was finished in 2000 and now contains around 7.5 GBytes of data. For NAVTEQ alone, over 500 field employees are working worldwide on data acquisition and maintenance [12].

The operators of purely commercial networks, such as airlines or public transport systems, are – of course – inclined to inform customers as optimally as possible about their services. Not all commercial providers are doing this equally though, public providers even less so. And, with the few that already provide good services in this respect, there is very little interaction between different services. They are mostly incompatible, either technically or by design. Interaction occurs only in those cases when the networks are complementary in nature – such as EasyJet offering train tickets for the Stansted Express from London Stansted airport to the centre of London, or hotel bookings which can be made in connection with a flight booking. Apart from these exceptions, those who own the most detailed data about infrastructures are generally not the first in line to sell their information or to provide a service of some kind.
The consequences for our geospatial world model are

- it will never be possible to have centralised access to a complete world model. Instead, the data will be distributed and only accessible through particular web services;
- the web services will not reveal data in a way that the whole database can be reconstructed by suitable sequences of queries. For example, if the web service provides route planning then the routes need to be described without detailed reference to the underlying road or transport network.

The first point requires an architecture where there is only a central coordinator of the world model, but the details of the model are hidden behind the interfaces of the various providers. This requires a quite complicated architecture, but it offers the possibility to change and extend the world model dynamically by linking new servers into the network.

### 3.1 Existing Approaches

Geospatial reasoning is a rather broad notion that has been looked at from various angles from within computer science and AI.

On the very concrete side there are the Geographic Information Systems (GIS), i.e. databases and algorithms which deal with the representation and use of concrete geographical data, road maps, land coverage etc.

'Shortest path' algorithms have been developed to solve the path planning problems, for example in transportation networks. The path planning problem in a concrete 2- or 3-D environment is one of the robot navigation problems, and there are a number of more or less practically useful algorithms to solve it [13].

Shortest path algorithms typically do not take into account context information about the traveller, e.g. if the traveller has a car available, or if he depends on public transport systems. One way to use context information in a shortest path algorithm is to construct a problem-specific graph so that, for example, if the traveller has a bicycle, the system might first construct a graph consisting of paths and roads, together with those railway and bus lines where a bicycle can be taken on board.

GIS techniques depend on the availability of concrete coordinates. If coordinates are not available, symbolic data representation and reasoning is necessary. One of the symbolic locational reasoning systems is the 'region connection calculus' (RCC8, [14]). It generalises the ideas of Allen's interval calculus from one to two dimensions. RCC8 provides basic relations between two-dimensional areas and has rules for reasoning with the relations.

A very general knowledge representation and reasoning technique are the Description Logics [2], with OWL as its WWW version [3]. In Description Logics one can define 'concepts', corresponding to sets of objects, and one can relate individuals to the concepts. The formula (1) is an example of a concept definition in a Description Logic.

Planning algorithms, originally developed within AI. [15] constitute one particular class of shortest path algorithms that can be handled very efficiently by precompiling an axiomatic problem representation into a graph. Certainly, route planning services can be regarded from this perspective.

Yet route planning services of different kinds will need to present the results of planning to users. The required style of presentation can vary enormously, both in terms of detail, and also in terms of modality (visual, verbal, audio, multi-modal).

One of the advantages of using graph structures as the basis of planning is that the output of a planning process is itself a graph - of a particular kind, with a formal structure that acts as a point of departure for a wide variety of different presentation styles.

Such variety needs to be anticipated to accommodate the unforeseeable nature of the environment under which the information might need to be accessed. This is particularly the case for the Semantic Web. For example, a user planning a trip from an office desk might profit from a presentation employing high resolution graphics and audio; a mobile user driving a car might avoid visual distractions by requesting spoken verbal description; a tourist on foot with a mobile phone might well prefer a low resolution sketch of the route through the city.

All these different presentation techniques can be based upon the same, underlying abstract plan structure by relatively straightforward generation techniques as illustrated by Rosner and Mizzi [16] and Rosner and Scicluna [17] which respectively deal with the presentation of natural verbal and visual instructions.

The reason this is possible is because there is a kind of isomorphism between the plan structure, and the elements out of which the presented description is based whether this be verbal, visual, or a mixture of the two.

## 4 Towards a Geospatial World Model

The examples in the introduction show that "geospatial reasoning" is very heterogeneous. Therefore we tried to develop a unified view of the area, which allows one to incorporate the various techniques and results in a single system.

### 4.1 Graphs, Graph Transformations and Ontologies

The basis of the unified view is the observation that in most of the approaches the data can be represented as graphs, and that there are close connections between the different types of graphs. We illustrate this observation with some examples.

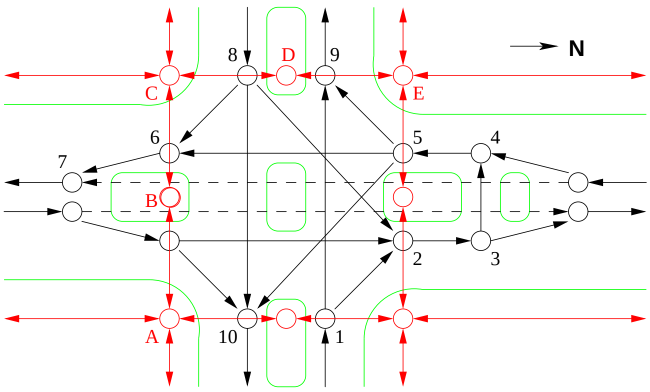



**Fig. 1.** Road Crossing: High detail

*Example 7 (Road Crossings).* Figure 1 shows a detailed representation of an intersection of two streets, including an underpass (dashed lines) and pedestrian pathways (shown in red). This graph is suitable for guiding an autonomous vehicle through the area of the crossing. A simplified version of this crossing is shown in figure 2. It contains enough information for a standard navigation system.

Finally, one can collapse the whole road crossing into a single node of the road network as seen in figure 3. This is sufficient for path planning on a larger scale.

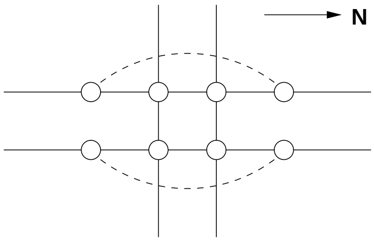



**Fig. 2.** Road Crossing: Medium detail

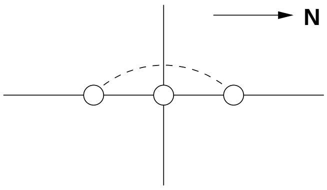



**Fig. 3.** Road Crossing: Low detail

In all three pictures we see the same road crossing, but on different level of detail. We are working at a language for describing how to generate the graphs with less detail from the graphs with more detail.

Different levels of detail are also pertinent to the problem of presenting solutions to geospatial planning problems in a way that is sensitive to the particular situation of the user and the resolution capabilites of the display device at hand. Rosner and Scicluna [17] discuss and implement the use of graph-reduction algorithms for simplifying the data at hand for efficient communication of information.

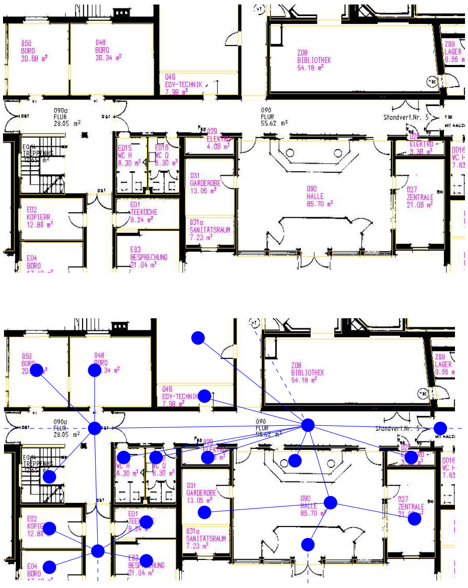
■

*Example 8 (Floor Plans).* Indoor navigation of autonomous vehicles requires a detailed floor plan, as shown in figure (1) of figure 4. In order to plan a way from, say, the entrance of the building to a particular office, such a detailed floor plan is not necessary. A simplified net plan, such as shown in picture (2) of figure 4 is much more suitable for this purpose. The simplified plan can be generated from the detailed floor plan. The convenient similarities between the examples 7 and 8, which present very different situations, are by design.

Finally, one can collapse the whole building to a single node in a bigger city map. The node is sufficient for planning a path through the city to this building. ■

*Example 9 (Symbolic Data Representation).* This example shows the transition from GIS style data representation to a pure symbolic knowledge representation.

The left hand side of figure 5 shows the boundaries of two of the German states, and some cities. The boundaries can be represented as polygons, and these are again just graphs. In the right picture the polygons are collapsed into single nodes of a graph.



(1)

(2)

**Fig. 4.** Plain Floor Plan without and with Network Overlay

The relation ‘polygon A is contained in polygon B’ is turned into an NTTP edge (Non Tangential Proper Part) of the new graph. The relation ‘polygon A touches polygon B’ is turned into an EC edge (Externally Connected) of the new graph. ■

The examples illustrate a number of observations

1. There is a hierarchy of graphs. At the lowest level there are graphs with the concrete geographical details which are necessary for, say, guiding autonomous vehicles. At the highest level there are graphs which represent logical relations between entities.
2. There are correlations between the nodes and edges of the graphs at different levels of the hierarchy. These need not be a one to one correspondence. Usually a whole subgraph of a lower level graph corresponds to a single node or edge of the higher level graph. A typical example is the representation of the city of Munich in Example 9, as a polygon in the left hand graph and as a single node in the right hand graph.
3. A transition from a lower level graph to a higher level graph can be facilitated by identifying specific structures in the lower level graph, and transforming them into structures of the higher level graph with the same meaning. In example 7 this structure is a road crossing. In example 8 these structures are floors, doors, rooms etc. In example 9 these are cities, states etc.
   These structures are in general part of an *ontology*. In parallel with the development of the graphs, we therefore need to develop the corresponding ontologies. The elements of the ontology are the anchor points for controlling the graph transformations and for choosing suitable graphs to solve a given problem.

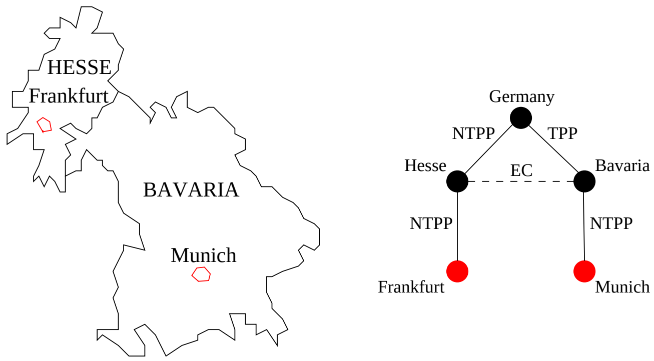



**Fig. 5.** Symbolic Data Representation

4. It is in general not a good idea to put all information into one single graph, even if it is information of the same level of detail. In a typical city we have, for example, a road map as a graph, the bus lines as a graph, the underground lines as a graph etc. We therefore need to consider collections of graphs with transition links between the graphs. Typical transition links between a road map and an underground map are the underground stations. The transition links, can, however, be little graphs themselves, for example the network of corridors and stairs in a big underground station.
5. The graphs at the higher levels of the hierarchy can and should usually be extended with additional information which is not represented in the lower level graphs. For example, the graph in example 9 with the symbolic information about cities and states can easily be extend by adding further cities and states.

### 4.2 A Road Map for the Development of Hierarchical Graphs

One of the most important goals is the development of a technology of 'geospatial' knowledge representation with hierarchies of graphs. The hierarchy connects the coordinate based GIS like information processing with the logic based symbolic reasoning. The following steps are necessary to achieve this goal.

**Step 1**: Unified Representation of Graphs.
The structures at the different levels of the hierarchy are all graphs. Therefore there should be a unified representation of these graphs. The graphs need, however, be represented in different forms.

– We need a persistent representation of graphs which can be stored in files or databases.
– We need an in-memory representation of the graphs with a well defined application programming interface, probably similar to the DOM structures of XML data.
– We also need geometric representations of the graphs which can be used to display the graphs on the screen. As long as the nodes of the graph have coordinates, this is

not a big problem. Graphs at the symbolic level of the hierarchy usually don't have coordinates. Fortunately there are well developed graph layout algorithms which we can use here.

Since graphs at different levels of the hierarchy can represent the same objects, road crossings, for example, it is very important to maintain the links between the same objects in the different graphs. These links enable algorithms to choose the level of detail they need for doing their computations.

It must also be possible to use the transition links between different graphs of the same level to join several graphs into one graph. For example, a route planner for somebody without a car may need a combined graph of all public transport systems.

As mentioned above, it should be possible to add extra information to the graphs, which is not derivable from graphs at the lower levels. In order to do this, we need to develop an *editor* for the graphs.

**Step 2**: 'Geospatial' Ontology.
We need to develop an ontology of interesting structures which can occur within graphs (road crossings, roundabouts, floors, train stations etc.). Such an ontology would be the anchor point for various auxiliary structures and algorithms, in particular:

– patterns which allow one to identify the structure in a graph, a roundabout, for example;
– transformation algorithms which simplify the structures to generate the nodes and edges in the graphs at the higher levels of the hierarchy;
– transformation algorithms which generate a graphical or verbal representation of the structures on the screen.

The ontology will also be used to annotate the structures in the graphs.

**Step 3**: Ontology of Graph Types.
The graphs at the different levels of the hierarchy provide the data for solving different kinds of problem. We need to classify the graph types, such that it is possible to choose the right graph for a given problem.

**Step 4**: Ontology of Means of Transportation.
A graph for a railway network, for example, represents only routes, but not the characteristics of the trains which are used on these routes. It can, for example, be important to know, which trains can take a bicycle on board, or which trains have wireless LAN on board etc. Therefore we need to develop an ontology for the objects which are connected with the graphs. If the graphs represent transportation networks, this must be an ontology of the vehicles used on the network. If, on the other hand, the graph represents, for example, a local area computer network, it must be an ontology of the characteristics of the cables together with an ontology of the devices connected to the cables.

**Step 5**: Context Modelling.
In the introductory examples we showed that queries which require 'locational reasoning' need to take into account the context of the user. We must therefore develop a

formal model of the context. The context can, for example, be the current situation of a human user: whether he has a car or not, whether he has luggage or not, his age and sex, and many other factors.

**Step 6**: Customised Graph Construction.
As we have seen in the introduction, many 'locational reasoning' problems require the solution of shortest path problems in a graph. The concrete graph which is relevant for the given problem, may, however, not be one of the graphs which are permanently available. It may be a combination of subgraphs from different graphs, and the combination may be determined by the context of the problem. Therefore we need to develop mechanisms for determining and constructing for a given problem the right combination of subgraphs as the input to the relevant problem solving algorithm.

**Step 7**: The Main Problem Solvers.
Finally we need to adapt or develop the algorithms for solving the main problems. These range from 'shortest path in a graph' algorithms to logical calculi for reasoning with symbolic information. Fortunately most of these algorithms are well developed and can, hopefully, be taken off the shelf.

### 4.3 Distributed Geospatial Services

The practical constraints, i.e. that businesses, organisations or governments make access to their data difficult and harbour potentially commercial interests leads to the need for a distributed architecture. Each and every provider in this architecture offers geospatial data either directly or through a set of services, as described in the following paragraph.

Whenever there exists an infrastructure of some kind (see section 4.1 for some examples), a corresponding web information server provides either a set of *services* regarding the infrastructure, or at least grants access to the necessary data. By services, we mean the processing of data in form of the above mentioned representation of geospatial data as graphs. Typical processing can be partly based on shortest paths, nearest neighbours, etc. Furthermore, from a software engineering point of view, services can easily be developed as highly reusable components which can be integrated within one device as well interoperating components over a network of distributed systems on the web. A set of services might include the following:

- **Routing Service:** Within a single graph, provide a route from one node to another.
- **Connection Service:** Provide a set of other graphs, which can be accessed from a given graph, including transition nodes.
- **Listing Service:** Provide a list of nodes or edges.
- **Integrity Service:** Check for the existence of connections between nodes within one or more graphs; e.g. "is office 136 in this building?".

The reason for not providing data directly, but instead the above mentioned services, is data protection. Whenever a provider wants to protect their assets by not disclosing information, they still have the opportunity of providing above mentioned services. Considering the substantial efforts required for geospatial data modelling and acquisition, data protection is likely to remain a central requirement for the service-oriented view. The data that is returned as an answer to a query might be provided in some form that does not allow for reconstruction of the original data sets – or at least make this operation too cumbersome and therefore not economically worthwhile. In cases where the infrastructure is publicly accessible, such as a street or public transport network, the need for data protection might have less importance. From the user's point of view, there might be little difference between the two, because whether the services and data are operated and/or provided by the same party or not, is typically irrelevant.

The main incentives for any provider to offer either data or services or both are the following:

– **Increased Revenue:** The better the quality and accessibility of the services (or data) provided, the more customers are attracted. An airline or railway company which provides easy to use information services and comfortable booking services on the internet will have an advantage over competitors with lower quality services.
– **Increased Efficiency:** By controlling the information and/or services about a network, a provider can significantly influence the use of the network itself. In cases where no direct revenue is generated, because the use of the network itself is free of charge, this may be the most powerful incentive. There are numerous possibilities for example in load balancing or directing traffic. The government of a city for example has great interest in optimising traffic flow, which is increasingly difficult to achieve by static means (signage) only.
– **Increased Value:** The value of a network increases with the number of connections to other networks. The more possibilities there are of accessing for example an airport, the more travellers will be attracted by the services provided there. If the only possibility to get there is "by car", then quite a big percentage of passengers will stay away.

### 4.4 Data Exchange Languages

We mentioned already a very important point, data protection. The results of a query to a server must be such that the underlying data cannot be reconstructed. For a route planning service this means that the generated route must be represented in a language which does not refer directly to the underlying graph. Instead one must use more higher level instructions like "drive along the main street until the fourth traffic light" or "board the train in Piccadilly Station" or "climb the stairs up to the third level" etc. This exchange language for routes refers to concepts in an ontology of actions like "drive along", "board a train" or "climb the stairs" etc. The language must be able to represent routes in a way such that

– partial routes can be concatenated to form longer routes
– particular steps in a route can be refined. For example, a route can say "drive to the airport", "board the plane". A refinement might be "drive to the airport", "park in the garage", "go to the check-in counter", "go to the passport control", "go to the departure gate" and "board the plane".
– the route descriptions can be verbalised or visualised. Prototypes of a verbalisation module [16] and a visualisation module [17] have already been developed.

A route description or *plan* language is one of the data exchange languages, probably the most complicated one. Other services of the distributed world model will require other languages. The resulting plan itself is a formal structure that acts as a point of departure for a wide variety of different presentation styles.

## 5 Summary

One of the key features of the Semantic Web is that data on the web can be interpreted with respect to their meaning, their semantics. The meaning can be represented in various ways, as ontologies, as axioms in some logic, as rules in some rule language, and even with special purpose procedures. In this paper we considered the meaning of 'geospatial' notions. Examples are 'in Munich', 'between Munich and Frankfurt', 'along the highway', 'next to the shelf with the milk' etc. We argue that a suitable representation of the meaning of these notions requires the development of a geospatial world model. Such a model is essentially a complete representation of all the geographic facts and relations of the real world out there.

Most of the geographic facts are already 'computerised' in GIS databases. The problem is that most of them are owned by companies with primarily commercial interests. In this paper we presented a proposal for a geospatial world model which can be used as the basis for interpreting geospatial notions in the Semantic Web. The basis of the world model are hierarchies of networks of graphs. At the bottom end of the hierarchy we have detailed maps of the geographic entities (road maps, underground maps, floor plans etc.) At the upper end we have purely symbolic representations of concepts and relations. The correlation between the different levels is by a, yet to be developed, language, which allows one to describe structures in the lower level graphs, which represent nodes or edges in the higher level graphs (road crossings, buildings, city boundaries etc.)

The fact that GIS data are usually not publicly available is taken into account by having a distrubuted architecture. A central server only coordinates the access to various other servers which provide access to their data. The response to such an access, however, must be a description of a problem solution which does not allow one to reconstruct the underlying data. Since many of the geospatial notions implicitly refer to route planning problems, a route planning service will be one of the important components of the geospatial world model. The result of a route planning request, however, must be described in a more abstract way than just as a sequence of edges in a graph. A "route markup language" is needed which, on the one hand, hides the underlying concrete data, and, on the other had, contains still enough information such that visualisation and verbalisation modules can generate useful presentations. Such a route markup language is only one, probably the most complicated, example for a data exchange language for the geospatial servers. Every class of queries to such a server needs an appropriate answer language.

The proposed road map for the development of hierarchical graphs and the concept of distributed data and services for geospatial applications for the Semantic Web pose an interesting challenge with the prospect of far greater integration than is offered on the web today.

## Acknowledgements

This research has been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net)


## References


1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **279** (2001) 35–43
2. F. Baader, D. Calvanese, D.M., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook. Cambridge University Press (2002)
3. W3C http://www.w3.org/TR/owl-guide: OWL Web Ontology Language. (2005)
4. Bry, F., Marchiori, M.: Ten Theses on Logic Languages for the Semantic Web. In: Proceedings of W3C Workshop on Rule Languages for Interoperability, Washington D.C., USA (27th–28th April 2005), W3C (2005)
5. Escrig, M., Toledo, F., Pobil, A.: An overview to qualitative spatial reasoning. In: Current Trends in Qualitative Reasoning and Applications, International Center for Numerical Methods in Engineering, Barcelona (1995) 43–60
6. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1** (1959) 269–271
7. Dijkstra, E.W.: Reflections on [6]. circulated privately (1982)
8. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach (2nd Edition). Prentice Hall (2002)
9. Ginsberg, M.: Essentials of artificial intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1994)
10. NAVTEQ http://www.navteq.com: Provider of digital map data. (2005)
11. Tele Atlas http://www.teleatlas.com: Provider of digital map data. (2005)
12. Grande, A., Kraus, B., Wiegand, D.: Standortbestimmung. c't, Magazin für Computertechnik (#10 / 2004) 84–89 (in German).
13. Gupta, K., del Pobil, A.P., eds.: Practical Motion Planning in Robotics. Wiley (1998) ISBN: 0-471-98163-X.
14. Cohn, A.G., Bennett, B., Gooday, J., Gotts, N.M.: Qualitative spatial representation and reasoning with the region connection calculus. GeoInformatica **1** (1997) 275–316
15. Fikes, R., Nilsson, N.J.: Strips: A new approach to the application of theorem proving to problem solving. Artif. Intell. **2** (1971) 189–208
16. Mizzi, D.: A Mobile Navigational Assistance System Using Natural Language Generation. BSc FYP, Department of Computer Science & A.I., University of Malta, Malta (2004)
17. Scicluna, C.: Multimodal & Adaptive Navigational Advice Generation. BSc FYP, Department of Computer Science & A.I., University of Malta, Malta (2005)