207

A CONCEPTUAL NAVIGABLE DATABASE MODEL FOR INTELLIGENT VEHICLE HIGHWAY SYSTEMS

Jonathan Gottsegen (JGotts@geog.ucsb.edu)
Michael Goodchild (Good@geog.ucsb.edu)
Richard Church (Church@geog.ucsb.edu)
National Center for Geographic Information and Analysis
University of California, Santa Barbara
Santa Barbara, CA 93106–4060

ABSTRACT

This paper describes research on appropriate data models for a navigable database to support Intelligent Vehicle Highway Systems (IVHS). This research has two main concentrations. The first is the most efficient conceptual model of a street network for the purposes of IVHS, and the second is an effective method for including representation of road lanes in the database. The first issue compares the planar model to a non–planar alternative. The second issue is important for the California transportation department who has funded the research. Because an accurate representation of lane geometry is not possible, the research proceeds with methods for describing lanes within the attribute data. This lane structure must support several types of information about the lanes including: data about the traffic flow in the lanes, possible turn directions from each lane, and connectivity between lanes at street intersections. The ongoing research is currently developing a prototype of the planar model with lane information to determine the effectiveness of the structure developed for storing lane information. If funding allows, the non–planar model will also be tested through a prototype project. This paper discusses the planar and non–planar models of the road network generally and the planar model in more detail. It also describes potential solutions to support lanes in the database. It concludes with a brief description of the prototype project.

INTRODUCTION

Intelligent Vehicle Highway Systems (IVHS) are being considered internationally as a means of managing traffic flow. The basis of this technology is a spatial database to support the applications on which IVHS are built. While there is considerable work progressing regarding technical aspects of IVHS, such as the most effective technology for transmitting positional or traffic flow information, there is little exploration of the most appropriate structure of the database. Even the discussions of a unique nationwide link identification system presume a data model rather than starting by deriving the most efficient data model and then uniquely identifying each element in the model.

The National Center for Geographic Information and Analysis at the University of California, Santa Barbara has been involved in a research project that is addressing the question of the best data model for IVHS. It has two components. The first is an investigation of the linear data model that most efficiently supports IVHS applications. There are two general approaches to this issue, the planar and non–planar model. The second component derives a structure for adding lane information to the database, and is included in response to comments that we have received from California's transportation department (CalTrans) expressing the desire for lane information and monitoring capabilities. Thus far, the research has reviewed the general issues surrounding various data models within the two general approaches and has researched the planar model in detail. It has produced a structure for lane data, and it is proceeding with the prototype database for the planar model.

This paper presents the results of this research to date. It proceeds with a brief review of the purpose and concepts of data models to provide context for the following general discussion of the planar and non–planar models. After the general discussion of the data models, it describes the planar model in more detail. It then reviews the issues surrounding the inclusion of lane information and briefly evaluates the methods for storing lane information. It concludes by reporting on the prototype project and its status. More detailed information about the research is available in the Progress Report and Final Report delivered to Caltrans from the NCGIA (Goodchild, et al., 1993; Goodchild, et al., 1994).

PURPOSE OF THE DATA MODEL

The data model provides the conceptual structure for the database. It is "a general description of specific sets of entities and the relationships between these sets of entities." (Peuquet, 1984) It provides a formal means of representing information. (Date, 1990) The data model is an abstraction of the real world, and it reflects decisions about what features and relationships are necessary to represent in a database. It must effectively replicate the way that the users of the database conceptualize the road network. In this case, the database and data model refer to the geographic database that provides the data necessary for routing and other important tasks in IVHS. So the data model must address how to represent spatial entities, e.g. roads, and relationships between them, and how to relate non–spatial attribute information to the spatial objects.

There are several levels of data modelling. They are external, conceptual, logical, and internal. (Laurini, 1992) The crux of developing a model is effectively mapping the features from one level onto the set of elements available in the level below it. This process is one of abstraction and symbolization so that the fewer elements maintained by the lower levels of models can encode the more varied features present in the levels above. (Milne, et al.; 1993) This paper is concerned with the external and conceptual level models. These are the levels that deal with how people view the phenomena being modelled.

There are many possible ways to structure geographic data and many data models possible. The planar data model is currently one of the most commonly accepted models because many GIS software packages and agencies that produce geographic data have employed it. One reason for this popularity is that the planar model is the least complex and most efficient model for databases of areal features. This model has certain advantages of simplicity resulting in its widespread use for certain analytical capabilities, but it has weaknesses as well. The non–planar model is not as prevalent, but certain other models possess some aspects of non–planarity such as a CAD structure, dynamic segmentation or some "feature oriented" systems.

DESCRIPTION OF PLANAR AND NON–PLANAR MODELS

The data models considered for this research are based on a single line (center–line) representation of roads for several reasons. First, network analysis routines, such as routing, are most effectively supported by the center–line road representation. Second, the center–line representation is more efficient for associating attributes with the single linear element, for deriving linear information such as distance, and for determining connections between roads from intersections of lines. Third, representing roads with a width, i.e. as double–lines, requires a scale of data compilation which implies a high degree of locational accuracy that is not financially feasible within the scope of this project.

The planar data model is derived from the concept of a planar graph. In this model, lines cannot cross without creating an intersection with an associated node. The crossing lines are therefore split into several individual links. The planar network is composed of two basic elements, links and nodes. Nodes terminate links, links always join two nodes, and links that terminate at the same node can be considered connected. Implementations of the model often store the links as independent elements with no intelligence regarding relationships to other links although it is technically possible to establish pointers that encode these relationships. This model is inherently two dimensional. It does not encode elevation differences between elements because it forces intersections between lines. Therefore, it does not implicitly store grade separations or physical restrictions from turning from one road onto another. If two links are connected, the model inherently allows turns from one to the other. Turntable types of enhancements that explicitly store turn restrictions are necessary.

There are several prominent examples of planar data models in commercial software and available data sets. One of the most commonly uses examples is the U.S. Census Bureau's TIGER files. The data in these files are structured as described above, and each type of element is named based on its dimensionality. The Census Bureau calls nodes 0–cells, links 1–cells and areas or polygons 2–cells. In addition, they have established a set of rules that define the relationships between these classes of elements to preserve the integrity of the database (Marx, 1986)

The planar model is most applicable in urban settings for several reasons. Most important among them is that geometric road intersections in cities generally are topological intersections as well. That is, they are actually points of connection between the roads where one may physically turn from one road onto the other.

The planar representation of less rectangular rural road networks or road networks with grade separations is not as effective. For example, the planar model of highway overpasses would split the highway into separate links at the overpass. If the exit and entrance ramps are included in the database, then the points at which they actually intersect with the linear representation of the highway would also require splitting the highway into separate links. In addition, there are no logical constraints to any turns at the nodes created by these intersections, so such restrictions must be encoded explicitly in an additional table.

An alternative to the planar data model is a non–planar model. The most significant difference exhibited by the non–planar model is that lines can cross without creating an intersection. The non–planarity of this model adds an implied third dimension in that if two lines cross without intersecting, they may be at different elevations, i.e. they are grade separated. A non–planar model introduces other benefits and efficiencies as well. The non–planarity makes maintaining integrity of data across intersection easier because the road intersection does not have to split the linear element into two separate links. They include more efficient geocoding and better compatibility with the general conceptual view of roads, although there may be an exception to this cognitive benefit. In urban areas, it is possible that people do consider roads in block segments. In this case the planar model would be most appropriate.

DETAILS OF THE PLANAR MODEL

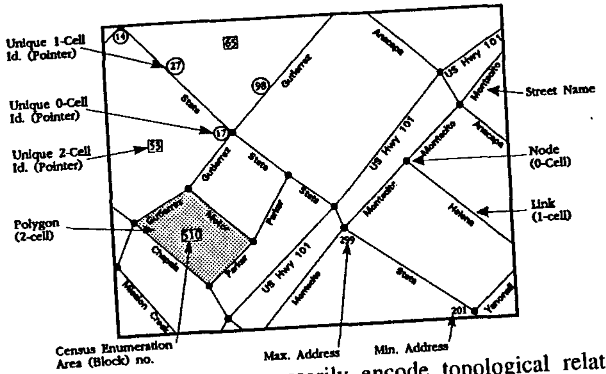
As mentioned, the planar model is constructed from links and nodes. They each have individual functions as well as prescribed relationships to each other. Links are used to represent any linear map feature, and nodes exist where links begin, end or cross which in the planar model implies the beginning and end of the links.

Links in the planar model represent road segments from one intersection to the next. One critical characteristic of links in the planar model is that they are completely independent of each other. Each segment of State Street does not "know" that is a part of the same street as the link connected to it. While these implicit relationships may be derived from the data in the model, such independence makes manipulation and analysis of the entire entity State Street less efficient. In addition, it necessitates redundancy in the attribute data and requires more explicit procedures to maintain integrity of the data over the entire road length.

Links can serve functions other than the obvious physical role of representing individual streets. For example, where a concrete barrier restricts left turns from a road, a separate link may be used for each direction of the road so that destinations do not appear on the left side of the link. Thus no possible left turns from the link will appear as an option.

Nodes in the planar model function as terminating points for links. They represent intersections because if more than two links terminate at the same node, then they intersect and are connected. Similar to links, nodes provide a physical function when they represent actual intersections. More often, however, they serve as topological entities through their relationship to links. For instance, a node exists at an overpass; the road and the overpass do not create a real intersection in three–dimensional space, but the two–dimensional planar environment models this relationship with a node. Figure 1 shows a portion of the U.S. Census Bureau's TIGER file for Santa Barbara County. The TIGER file elements are labelled, but not every implementation of the planar model contains these elements.

Figure 1
TIGER Data Model



While the planar model may not necessarily encode topological relationships, i.e. characteristics that are invariant under scale changes and include spatial relationships between objects such as adjacency or connectedness, many of the most common implementations of the planar model in software and in data sets construct topology for the database. The topology adds intelligence that encodes which nodes terminate which links, and therefore the connectivity between the links. The topology is typically maintained through a series of tables, at least one for each type of entity and pointers between them. Each record in a table is an instance of the respective object type and it points to records in the tables for other related objects types.

Road Attributes in the Planar Model

The planar model has difficulty handling attribute information that refers to only parts of links. This often occurs in transportation applications. For example, the paving surface of a road may change along a link. This is the main motivation for the development of dynamic segmentation procedures for transportation GIS. The problem of attribute changes along a link creates an inconsistency between the basic object in the geometry of the planar model and the basic entity in the relational table. If an attribute changes along a link, this would require two or more instances of the attribute for the link, which would translate into two records in the attribute table for the link. The pure planar response to changes in attributes within a link is to split the link where the attribute changes. This would create two records in the table instead of one which would clean up the inconsistency between the attribute table and the geometry, but it would require reconstructing the topological relationships between the objects in the database. As databases become larger and more complex, this solution becomes increasingly less effective. In addition, solution clearly cannot handle continuous change along a link (actually continuous change poses a problem for both planar and non–planar models).

Functionality of the Planar Model

The applications that are often included under the purview of IVHS range from advanced transportation information systems (ATIS) to advanced vehicle control systems (AVCS). They depend on a variety of routines or tasks, the most significant of which is a collection of

The procedures include finding the best route from a given point to another specified point (route finding) and providing graphic or verbal directions (route guidance). Other tasks require locating points and events of interest or potential travel restrictions, such as accidents, and transmitting these locations to drivers or emergency response agencies (route information, route conditions); monitoring and transmitting road usage and congestion to manage traffic along a network; and rerouting a vehicle to a destination given an obstruction or increased congestion on a road or portion of a road network.

The main applications in this set are routing, including route finding and route guidance, and geocoding or address matching. Both of these applications serve as the foundation for more complex IVHS tasks. Both of them are also fairly well developed applications with considerable effort devoted to them over several years and mature robust algorithms for their calculation. Route guidance is simply a matter of translating the optimal path into turn directions, but it requires care in certain instances such as turning across lanes to enter the destination. The database must contain the detail to determine whether this is possible. The route guidance task is considerably more complicated if lane detail is introduced into the directions as is the goal of this research.

One of the most significant advantages of the planar model is that it is common and fairly simple. The relationship between the links and nodes is consistent within the conceptual planar model. That is, links that cross will always create an intersection with a node. This implementation of the model is more straightforward because every instance of an entity can be treated identically in constructing the data structure. The planar data model is also adequate for geocoding in urban areas where the address ranges exist and are consistent and linear interpolation along a street segment fairly accurately replicates the progression of addresses in a block. Geocoding is typically a difficult problem, however, because people often write addresses in different ways.

The disadvantages of the planar data model include some inefficiency in GIS routing implementations because of the model's planar structure. Since there are nodes at every link intersection and the data structure logically allows a turn at every node, it would be logically possible to turn in any direction at every street crossing, regardless of the physical separation. An additional turntable data structure in the data model is necessary to redress this problem, but the additional structure and the presence of superfluous nodes adds to the processing requirements for network analysis tasks. In addition to the processing overhead, the planar structure does not replicate how people conceptualize the road network. In non–urban areas, one usually does not think of roads as split into separate segments at each intersection, and one often associates attributes to long lengths of roads, not to a series of smaller segments.

LANES

In addition to the data model, the NCGIA was charged with determining the best model to support lane information in the database. The lane structure must store attribute information about the lanes, i.e. traffic flow and obstructions, and provide a basis for directions that specify turns into and out of lanes. As discussed in the introduction to the planar model, the database will not have the accuracy necessary to represent lanes in the geometry of the database. Rather, lane information will have to be stored in various tables and related to the links.

The problem posed when introducing lanes into any single line model for a road network is that the model requires a hierarchical relationship between links and lanes, as there are potentially many lanes in each link. This may be approached from two directions. First, the lanes may be considered attributes of the links, and information about the lanes can be stored in an attribute table of the links. Second, the lanes can be considered entities worth modeling themselves. In this case they would be stored in their own table, and their relationship to links would be stored as attributes of the lanes.

Use of Lane Information
There are many ways to integrate lanes into the model, but the most appropriate way will depend on the use of the lane information. There are four potential uses of lane information in IVHS: providing turn directions, directions for information on and avoidance of obstructions in specific lanes, monitoring traffic flow within lanes for modeling purposes, and, related to providing directions, representing beginning and ending points of lanes.

Providing turn directions is part of route guidance tasks in IVHS. The system has to determine from the database the presence of a turn lane in order to give a driver directions for changing lanes. In addition, the system must account for lanes beginning or terminating along a link to alert the driver of necessary lane changes. For these purposes, the appearance or disappearance of a lane should be recorded with its point along the link so the system can give directions at appropriate times. Turn lanes may not require such locational information because a direction to change into a turn lane before the turn is sufficient. In addition to these circumstances, the system must store whether a specific lane is direction restricted to alert a driver to get out of a lane if necessary.

Avoiding obstructions in specific lanes requires the ability to locate the obstruction in a link and give appropriate directions to avoid the obstruction. The directions might entail changing lanes at a certain point or avoiding the link entirely by rerouting the driver. The latter approach also depends on monitoring traffic flow along a link. The location of obstructions in a lane is important for public safety purposes as well, and would have to be related to some milepost or offset along a link.

Monitoring congestion in lanes provides input to detailed traffic engineering models that propagate traffic flow changes along a link and through the lanes. In this case, monitors will be at specific locations along the link and will be combined with obstruction information. As mentioned, the beginning and ending points of lanes are necessary for lane change directions. The beginning and ending points must be located along the link in order for the system to provide timely directions.

Representing Lanes
The general question of this project is how to store the information necessary for the uses enumerated above in the manner that most efficiently fits within the single–line representation of roads in either a planar or non–planar data model. This paper addresses only the planar model. Future research will concentrate on the non–planar model. The ultimate decision about which method is most appropriate will depend on the objectives for the database in terms of both the application objectives and the logistical objectives related to computer storage.

There are several considerations for including lanes in the model. They include the appearance and disappearance of lanes, connectivity between lanes both at turns and between parallel lanes in the same link, lateral and forward obstructions and restrictions to movement, and restrictions to turns from a link such as physical barriers in the center of the road.

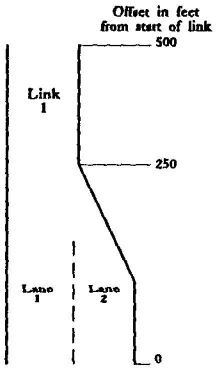
Planar Implementation. A pure planar implementation of lane information would split links each time the characteristics of the link change. Thus if the number of lanes along a link changed, the planar model would split the link at the point of change. In this model the lanes would be attributes of the link as would attributes of the individual lanes. Thus the attribute table for the link would have to contain a field for each lane attribute of interest for each lane. The number of attributes added would be the product of the lane attributes multiplied by the number of lanes.

The advantage of this approach is analogous to the advantage of the data model in general. It is conceptually simple. It is easy to conceive of separate records for each portion of a link that has a different number of lanes from the links that connect to it. However, several problems exist with this approach. First, reconstructing the geometry and topology when of lanes or attributes change in a lane along a link is extremely cumbersome. In fact, this task would make real time, or even timely, updates impossible. Second, this approach would increase the size of the database considerably for two reasons. First, the number of links in the geometry, and therefore the number of records in the database, would proliferate rapidly. Second, the number of additional fields required by this approach would be large and make the link attribute table extremely large as well. More important however, is that the number of lanes along the links in the database varies, but this approach would depend on fixed record lengths in the attribute table. This would consume significant space unnecessarily.

The other implementation strategy is similar to a dynamic segmentation approach, but it does not construct larger routes from the individual links. It uses offsets along the links to store the beginning and end points of the lanes. This method views the lanes for each link as independent entities to be modelled while the pure planar method views the lanes as attributes of the links. This strategy is actually a geocoding technique where, instead of using addresses along the link, the location is based on some mile offset from the start of the link or some percentage coverage of the link.

Another option deals with how the number of lanes is actually stored. This can be accomplished by storing the span of specific lanes or by storing the number of lanes as the phenomenon that occurs along a route (as shown in figure 2), i.e. storing the offsets of the individual lanes or the offsets of the two–lane portion of the road and then the one lane portion. The advantage of the second method over the first is that it also can reduce the data storage requirement. For example, using the start and end offset method rather than the length offset method, a link that changes from two lanes to one lane would require four pieces of data, two for the offsets of the two–lane portion and two for the one–lane portion. If each lane were stored with its offsets, this would require six pieces of information, four for the two–lane portion (two for each lane) and two for the one–lane portion. One disadvantage is that the actual start and end offsets of the specific lanes are implicit and must be derived by comparing the lanes to the offsets for the number of lanes. Another disadvantage that may be more significant is that the second method does not maintain which lanes continue along the entire link. The first method can store this by making the start and end offsets of specific lanes the beginning and ending of the link. The second method forces discontinuity in the lanes.

Figure 2
Options for Recording Change in Number of Lanes Along a Link



Option 1
Link Table

Link #	# Lanes	Start Offset	End Offset
1	2	0	250
1	1	250	500

Option 2
Link Table

Link #	# Lanes	Portion of Link Covered (feet)
1	2	250
1	1	250

Lane Connectivity. One of the suggested reasons for incorporating lanes into the data model is to support connections between lanes for route guidance and tracking traffic flow for monitoring purposes. There are a variety of means for accomplishing this, creating a spectrum of the amount of information related to driving behavior amount that is explicit. All of them require some structure analogous to a turntable that records connections between lanes at turns and can potentially store impedances for possible turns from each lane to every other lane. This report will describe three examples, for illustrative purposes. In addition, it is possible to distinguish logically between connections between lanes in the same link and between lanes in different links at turns. They will be treated separately in this section. Connectivity at turns will be discussed first, followed by parallel connectivity.

One possibility that will not be discussed is actually omitting information about lane connectivity. While there are arguments in favor of this, we assume here that such information is desired for various reasons. Among the potential reasons for requiring lane connectivity information are the ability to handle cases of connectivity of special lanes such as bus lanes or carpool lanes or where several lanes converge on a highway but do not merge, i.e. the far left lane from one road may no longer be the far left lane on the next road.

The minimum information about turn possibilities from lanes that could be stored would be directional restrictions on lanes, e.g. left turn only. These must still be included in the database. One possible strategy for this is to use an attribute or set of attributes to indicate the possible turns from a lane. The attributes may be binary (whether a turn is possible in a certain direction) or they may contain numeric values for the allowable turns. The first technique is more flexible because it allows the precise turn to be stored rather than imposing artificially discrete turns on the physical configuration. However, both of these strategies suffer from the problem of fixed length records trying to model a variable number of possibilities. That is, there may be numerous possible turns or only one possible turn, but the table storing the valid turns will have a given number of turn restriction attributes for each record or lane.

Another possibility for this level of lane connectivity is a turntable where each record represents a lane and one possible turn onto a link. This is an extension to the turntable structure to store turn impedance and restriction data for links. Restrictions can be explicitly indicated in this table or may be implied by omitted records. Either way, the turn data will match the actual turn configuration more closely and additional information may be maintained for specific turns from each lane. Besides the obvious larger data storage requirements, the disadvantage of this approach is that it still does not handle special use lanes well.

The most detailed and demanding approach to lane connectivity is actually storing the topology of lanes. That is, the connection of each lane to other lanes may be included with a more expanded turntable. In this case each record in the table represents a turn from a lane into a specific lane on another link. Clearly the turntable will expand rapidly. This approach allows the most flexibility because it contains the most detailed data. Its disadvantage is the size of the data structure required and the management overhead involved.

Turns from a Link or Lane. A large problem in providing route guidance on a link basis is providing the intelligence in the database so it knows whether a landmark along the link is accessible when traversing the link in a given direction. On a planar model without lanes, this is a significant problem. When lanes are included, it expands the problem. The lane database must reflect not only when turns across other lanes of traffic are restricted, but also when lane changes between lanes in the same direction are not possible. We have called this set of restrictions lateral obstructions.

In the case of modelling lateral obstructions from a link, the main question is whether the restrictions should be reflected in the geometry of the database or in attributes of the links. Restrictions represented in the geometry would use a link for each separate parallel length of road segment. For example, it is possible to model a street with a boulevard–like divider with a different link in each direction of travel. The links act as two adjacent streets with no inter–connectivity. In addition, all of the landmarks will be on the right side of each link, because none will be located in the middle, i.e. on the island. Therefore, the landmarks will not be accessible by a left turn from the link on the other side of the divider because they will not be adjacent to it. The same logic applies to roads that are divided by a cement barrier. They are the same road, but the two sides of the road are not accessible, and one cannot turn left along the links. In many cases, the distance between the two road directions is not large enough to be legitimately within the positional accuracy of the road database. Gaps in the divider that allow left turns can be represented by short links between the two road segment links.

Another way to model lateral restrictions is to include an attribute in the link database that indicates whether a left turn is possible or not. This reduces the number of spatial objects in the database, but it cannot handle gaps in the lateral restrictions as easily. These are point locations where the restriction attribute changes and then returns to the original value. This raises the same issues as that of modelling the presence of lanes, or any attribute that changes along a link. Since this problem is analogous to the lane problem, a similar solution is possible. Lateral restrictions may be an entity modelled in itself and located along the links as offsets from the start node of the link. This would allow infinite changes in the restriction along the link.

Both of these methods must deal with barriers disappearing part of the way along the link. For example, it is common for small concrete islands to exist between the opposing directions of a road near busy intersections in order to prevent left turns that conflict with traffic at the intersection. In addition, both methods must support the routing applications so that a driver is guided onto the correct side of the road in order to arrive at the desired destination. Each method of representing lateral restrictions is better in one of these special requirements. The attribute method is more flexible in handling changes in the restriction along the link. The separate link method would require the two links to be merged into one which might be confusing to the user of the database. On the other hand, the separate links are the more efficient means for supporting routing algorithms, because they may derive their turn behavior information primarily from the geometry. One consideration in weighing these alternatives is the nature of the restriction. If it is a regulatory or less permanent physical restriction, modelling it geometrically may require too much restructuring of the database should the restriction change.

Lateral obstructions may also be important at the lane level. Clearly, it is important to know whether left turns from a central lane are restricted as on roads with central barriers, but this information may be inherited from the link information. It may also be illegal or impossible to change between parallel lanes in a road segment, for example, carpool lanes or temporary construction lanes that restrict entrance from other lanes. It is undesirable to model the lanes with separate geometry, so the restriction attribute is the most appropriate method. However, to reduce the number of records in a table that stores these lateral restrictions, it is possible to assume the interchange between the lanes is valid unless a restriction record exists. This record would indicate the start and end offsets of the restriction.

This type of method overcomes the analogous problem in electrical or phone cable network models. Some have compared the lane connectivity problem to modelling connections between wires in an electrical trunk cable. Electrical network models typically separate the individual wires in a trunk cable with the geometry of the representation. Wire crossings can be indicated in the geometry as well. This is not possible in road network models however, because parallel lanes are infinitely connected because there are an infinite number of points along a line. Therefore the continuity of parallel lane connections is broken where the access between the lanes changes. This discrete point can be stored in the attribute table for the lanes.

THE PROTOTYPE

The purpose of the prototype is to demonstrate the effectiveness of the planar and non–planar models to support IVHS functions. It will illustrate the feasibility of implementing the solutions to the special issues discussed in this paper related to building a database for IVHS. It will construct a planar database for a portion of the City of Santa Barbara using data provided by ETAK, Inc. and Navigation Technologies, Inc (Navtech).

Constructing the prototype will include importing the data sets received from the two vendors, constructing the structures necessary to support lane information and other IVHS needs, and implementing rudimentary versions of IVHS applications. The resulting product will be a database with detailed information and a small set of routing and other routines to simulate IVHS operations.

To date, we have begun implementing the prototype for the planar model only. We have read in the data from both companies and populated the lane database for the Navtech data. One of the interesting necessary steps in using both data sets was understanding and comparing the data models used by each company because they differed slightly. Examples include the data model used to represent certain instances in the road network, the presence of turn restrictions and the method for storing them, and the exact scheme for storing addresses along links.

The issues that the prototype will address will include an enumeration and analysis of road and turn configurations that require special attention. Such pathological cases include turn lanes that do not intersect with another link; bus, bike or carpool lanes; or roads that are geometrically discontinuous but continue in name.

CONCLUSION

This research focuses on the data model that most efficiently supports IVHS. The data model considerations are often not addressed in the discussion of IVHS because it is assumed that certain databases will be used. However, this research proposes two alterations to the traditional data model, a non–planar model and the addition of lane information. While the non–planar model has some advantages on paper, it is not clear whether those advantages offset the possible disadvantages without testing the model. One of the clear disadvantages is that much data that exists follows the planar mode. The lane information introduces considerable overhead, but it may provide extremely useful information that has not been considered previously.

REFERENCES

Date, C.J. (1990) *An Introduction to Database Systems.* Addison–Wesley; Reading, MA.

Federal Geodetic Control Subcommittee (1989) *Multipurpose Land Information Systems: The Guidebook.* Federal Geodetic Control Subcommittee; Washington, D.C.

Goodchild, Michael F.; et al. (1993) *NCGIA Progress Report; CalTrans Agreement 65T155, MOU1.* University of California at Santa Barbara.

Goodchild, Michael F.; et al. (1994) *NCGIA Final Report; CalTrans Agreement 65T155, MOU1.* University of California at Santa Barbara.

Laurini, Robert and Derek Thompson (1992) *Fundamentals of Spatial Information Systems.* Academic Press; San Diego, CA.

Marx, Robert (1986) The TIGER System: Automating the Geographic Structure of the United States Census. *Government Publications Review,* 13, pp. 181–201.

Milne, Peter; et al. (1993) Geographical Object–Oriented Databases–a case study. *International Journal of Geographical Information Systems,* 7, 39–55.

Peuquet, Donna J. (1984) A Conceptual Framework and Comparison of Spatial Data Models. *Cartographica,* 21, pp. 66–113.