NCGIA

National Center for Geographic Information and Analysis

Final Report

Caltrans Agreement 65TI55
MOU 1

Richard Church
Danette Coughlan
Thomas Cova
Michael Goodchild
Jonathan Gottsegen
David Lemberg

Caltrans Testbed Center for Interoperability
Ramez Gerges, Caltrans Program Manager

Technical Report 94-6

October 1994

Simonett Center for Spatial Analysis
University of California
35 10 Phelps Hall
Santa Barbara, CA 93106-4060
Office (805) 893-8224
Fax (805) 893-8617
ncgia@ncgia.ucsb.edu

State University of New York
301 Wilkeson Quad, Box 610023
Buffalo NY 14261-0001
Office (716) 645-2545
Fax (716) 645-5957
ncgia@ubvms.cc.buffalo.edu

University of Maine
348 Boardman Hall
Orono ME 04469-5711
Office (207) 581-2149
Fax (207) 581-2206
ncgia@spatial.maine.edu

1. INTRODUCTION

A. Overview

This is the second and final report to Caltrans by the National Center for Geographic Information and Analysis at the University of California, Santa Barbara, under MOU 1 of Agreement 65TI55. It should be read in conjunction with the Progress Report (NCGIA, 1993) submitted in August, 1993.

The purpose of the project is to develop the concepts, methods, and techniques of navigable map databases for IVHS (intelligent vehicle highway systems). Most of the anticipated functions of IVHS rely on the existence of an accurate, dependable map database, with sufficient information on network connectivity and driving conditions to allow solution of such problems as determination of the shortest path between a given origin and destination. However, there are many possible design options for such a navigable map database, and its relationship to existing technologies such as geographic information systems (GIS) and the global positioning system (GPS) is far from clear.

The NCGIA/UCSB project has explored three major aspects of navigable map databases: the elements to be stored in them, appropriate data models, and issues of database distribution. These three aspects form the subjects of the three major sections of this report. Section 11 describes the major functions of IVHS, and their requirements, if any, for navigable map databases. Section III discusses alternative data models, including the traditional planar link/node model and newer, more powerful models. Section IV examines the design of a distributed navigable map database, and the capabilities of current GIS and database management products in this area.

Much of the previous literature on map databases for transportation has focused on the conversion of standard road and street map information to digital form. Such information represents the transportation network as a set of road and street center lines, intersecting at nodes. There is frequently ambiguity where lines cross, as it may not be possible to determine if the crossing is at grade, and thus whether interchange is possible. This NCGIA/UCSB research has stressed the need to consider non-planar models, in which the network is embedded in a three-dimensional space with no ambiguity at crossings; and also the need to include connectivity not merely between links in the network, but between specific lanes. Only lane connectivity can provide a fall, robust representation of such constraints as time-dependent turn restrictions. Thus this research departs from previous work in this area in several significant respects.

The Testbed Center for Interoperability (TCFI) coordinated and facilitated the flow of information between the NCGIA/‘UCSB and other organizations involved in IVHS activities. TCFI has provided numerous inputs to this report, especially in the sections on Applications and Elements, and Distributed Database Issues, and in the Appendix. TCFI has provided an interface to Caltrans, FHWA’s IVHS System Architecture, SAE, Oak Ridge National Laboratory, and the ATMS Testbed, and has coordinated access to ETAK and Navigation Technologies map databases. ETAK and Navigation Technologies have kindly provided access to their data for Santa Barbara for test purposes. Cecil Goodwin of the University of Tennessee Energy, Environment and Resources Center provided helpful comments on an earlier draft. The National Center for Geographic Information and Analysis is supported by the National Science Foundation, grant SBR 8810917.

The remaining subsections of this introduction describe the relationship between this project and the broader developments of GIS for transportation; and between this project and national efforts to define a standard link ID for IVHS. An Appendix to the report describes a subsidiary study of the use of differential GPS (DGPS) in IVHS.

B. GIS for Transportation

A recent report describes the use of geographic information systems for a variety of transportation applications, and includes a survey of GIS use among state departments of transportation (Vonderohe et al., 1993; see also the digest of results of this project, NCHRP, 1993). The report includes a full and comprehensive discussion of GIS data models, functionality, and trends, and provides a useful guide to a transportation agency considering GIS acquisition. It would form a very useful introduction to GIS for readers of this report who may be unfamiliar with GIS principles and applications in transportation.

C. Relationship to the National Spatial Database and Link Identification for IVHS Project

The ma or thrust of the National Spatial Database and Link Identification for IVHS Project is "to develop a method and strategy for testing and implementing a national spatial database and link referencing system". The reasoning behind the effort is that for the various applications of IVHS, particularly Advanced Traveler Information Systems, a nationwide transportation database is needed to ensure the interoperability of systems and databases across spatial boundaries (FHWA, 1993). In addition a standard will allow a quality control level between separate vendors of road network database products and allow users to select data from multiple sources. Along with the digital map database requirement is a requirement for a common link referencing system. This system will allow agencies to easily transmit link identification and location data as a unique identifier that can be used by any road database from any vendor.

The National Spatial Database and Link Identification for IVHS Project will include the following tasks:

1. Develop Functional Requirements for Map Databases
2. Develop Technical Requirements for Map Databases
3. Identify Approaches for Developing Databases
4. Review GDF and SDTS
5. Define a Transportation Link Referencing System
6. Make recommendations to the FHWA
7. Test and Evaluate the Network Referencing System

Much of this effort parallels or interfaces with the efforts of the Caltrans Navigable Database Project and with other efforts in the industry. Tasks One and Two - the functional and technical requirements for map databases - lead from work already conducted by the DOT, IVHS America, and the Society of Automotive Engineers to develop user service requirements for IVHS. The U.S. Department of Transportation Program Plan for the intelligent Vehicle Highway Systems (IVHS) Program (July 1993) and the Mission Statement (10/93) detail 27 IVHS Applications (now 28 User Services) and their functional and technical requirements, but do not address in depth the technical specifications of the underlying map databases.

The third task of identifying approaches to developing databases is one section where there should be real communication between the Caltrans Project and the Federal Project. Although most if not all of the available map databases reviewed by Gordon, Goodwin, and Xiong (1994) use planar data models, the non-planar models discussed in this report can offer significant advantages. The work of the Caltrans project should be considered in this area. in addition, Siegel and Okunieff (1994) have recently outlined several of the alternative schemes for location referencing, and associated data models.

The other area in which the data models of the Caltrans Project should be integrated is in the development and testing of link identification systems. Again, much of the previous effort in this direction seems to be based on planar views of the road network. At the Society of Automotive Engineers International Traveller Information Standards Committee Location Coding Workshop in San Francisco in July, representatives from the FHWA, the SAE, and IVHS America discussed the requirements for Location Coding Standards. They included:

1. A Flexible Approach - addressing areas, lengths of routes, intersections, and other points and links. Ideally this should be a single system.

2. A Versatile Approach - usable by people, agencies, GPS, and navigation systems. The approach should b compatible with all potential users.

3. Multimodal - usable for automobiles, trucks, bus transit, rail transit and freight, airlines and ferry services.
4. Machine Processable - designed for automated searching and sorting.
5. Fast - designed for rapid access without sorting in common use.
6. Compact Codes - for efficient use of bandwidth and speed of transmission.
7, Non-Proprietary - giving compatibility between vendors.
8. Affordable - not too complex and not too huge.
9. Maintainable - able to accommodate changes without starting over.

The basic approaches to Location Coding included using Place Names, Intersecting Street Names, Lat/Long Coordinates, and Mileposts/Offsets. Each of these systems have advantages and disadvantages. Place Names are flexible and affordable and easily recognized by people, but difficult to process by machines. Intersecting Street Names are also easy to use by humans but are limited to urban areas with street names. They do not work for polygon and off road point locations. Lat/Long Coordinates are easily coded and provide a standard reference, but are not recognizable by most humans and require a very high level of precision on the map database. Milepost/Offsets are compact, public domain, agency friendly, but not people friendly.

Each of these methods should be analyzed for implementation on the various data models developed in the Caltrans Project. Place Names would be difficult to process under planar or non-planar models. Intersection Street Names are suitable to a planar implementation with nodes at each intersection, but would need special treatment in a non-planar system with intersection tables. Lat / Long Coordinates would need to be matched to nodes in a planar implementation and milepost in a non-planar implementation. Milepost/Offsets would be particularly effective in an non-planar implementation, but would involve a milepost identification on a planar system. Another location coding issue treated in the Caltrans Project is lane level location. Many IVHS applications use lane level information. The problem of link identification coding should also include some system of lane identification to be compatible with applications systems requirements.

II. APPLICATIONS AND ELEMENTS

A. Introduction

This section will center on the idea of a standard set of spatial location level data elements required for implementation of a navigable database for IVHS. This section will describe the 27 applications listed in the U.S. Department of Transportation Program Plan for the Intelligent Vehicle Highway Systems Program. Given this assortment of IVHS applications, it will not be possible (or advisable) to generate a list of standard data elements for each application. The data elements required for an IVHS application in a navigable database are dependent on the implementation requirements of each public or private agency. Each will be customized to the organization designing the systems.

What is required for interchange of data is a set of special data levels. These spatial data levels will be the "handles" on which data may be linked and accessed on any implementation of a navigable database for IVHS applications. They will be a locational foundation to attach data specific to applications ranging from traveller information systems to traffic control models. The requirements for the base spatial levels are independent of the data model, but the actual data elements (intersections, lanes, links, routes, points, polygons) will vary with the implementation of the models.

B. The Base Spatial Data Levels

This section will detail the requirements for the base spatial data levels for a navigable database for IVHS. These data levels follow the basic structure of the primitive vector object definitions of the Spatial Data Transfer Standard (SDTS) (Davis, George, and Marx, 1992). This structure allows the system designer the following advantages:

1. Interoperability - the ability to pass data between systems by adding fields or objects to common structures,

2. Easy implementation - can work on hierarchical, relational, or object oriented database platforms.

3. Flexibility - allows geometric location and topological relationships between point, line, and area objects.

The following table details the SDTS primitive vector objects. There are three dimensions of objects: zerodimensional (point level), one-dimensional (line level), and two-dimensional (area level). These definitions will be used to define a set of base spatial objects for use in an IVHS navigable database.

Table 1: SDTS Spatial Object Definitions

Zero-Dimensional Objects

Point		An object that specifies a single spatial location.
Point Subtypes:		
	Entity Point	A point used to identify the location of a feature such as a milepost, building, tollbooth, etc.
	Label Point	A point used to identify the location of text or symbol on a display.
	Area Point	A point representing an area for purposes of storing attribute information about the area - a centroid.
Node		An object that represents the junction of two or more links or chains, or the termination of a one-dimensional object.

One-Dimensional Objects

Line Segment		An object representing a straight line connecting two points or nodes.
String		An ordered sequence of connected nonbranching line segments (a string may intersect itself or other strings).
Arc		A curve that is defined by a mathematical function.
Link		A topological connection between two nodes.
Chain		A directed, nonbranching sequence of nonintersecting line segments and/or arcs bounded by nodes.
Chain Subtypes:		
	Complete Chain	A chain that explicitly references left and right polygons and beginning and ending nodes.

Area Chain	A chain that references left and right polygons, but not beginning and ending nodes.
Network Chain	A chain that references beginning and ending nodes, but not left and right polygons.

Ring	A sequence of nonintersecting chains, strings, and/or arcs that close to form the boundary of an area.

Ring Subtypes:

G-Ring	A ring created from strings and/or arcs.
GT-Ring	A ring created from complete and/or area chains.

Two-Dimensional Spatial Objects

Interior Area	An area not including its boundary
G-Polygon	A area consisting of an interior area, one out G-ring, and zero or more nonintersecting, non-nesting inner G-rings (may have enclaves).
GT-Polygon	An area that is an atomic two-dimensional component of one and only one two-dimensional manifold (no enclaves). A GT-polygon can be associated with its chains.
Pixel	A two-dimensional picture element that is the smallest nondivisible element of a digital image (display).
Grid Cell	A two-dimensional object that represents the smallest nondivisible element of a grid.

These objects apply well to the design of a navigable database. The definitions are general to reflect wider geographic data requirements than just navigation. There are some structures missing from the standard including intersections, routes, and lanes. There is also a bias in this standard toward a planar data model, since all of the definitions are in terms of links and nodes. The following spatial data objects can be implemented in planar or nonplanar data models.

A. Point Level Spatial Objects

- Intersections (can be implemented as nodes or milepost/offsets)

  An intersection is an object representing the junction of two or more street links or chains, or the termination of a street link or chain. An intersection differs from a node in that it may have multiple facets for each lane entering or departing the intersection, thus an intersection is a zerodimensional object with two-dimensional aspects. Intersections create the base structure for turntables.

- Link Points (can be implemented as nodes or milepost/offsets)

  Link points are points located between nodes on links. Links points may also be located on individual lanes.

- Offset Points (can be implemented as entity point or milepost/offsets)

  Offset point refer to entity point locations off the road network.

- Polygon Points (can be implemented as area points or milepost/offsets)

  Polygon points are area points used to store attribute information about a polygon.

B. Link Level Spatial Objects

- Links (can be implemented node to node or milepost/offset)

  A link is a topological connection between two nodes. This definition differs from the SDTS definition in that links in an IVHS navigable database may be implemented with lanes making a link a one-dimensional object with two dimensional aspects. Direction is optional with links.

- Lanes (can be implemented node to node or milepost/offset)

  Lanes are directed topological connections between two nodes. Lanes between nodes are bound into links. Direction is required for lanes.

- Routes (can be implemented as collections of links - node to node with waypoints or milepost/offset to milepost/offset with waypoints)

  A route is a network chain associated with a vehicle path.

C. Polygon Level Spatial Objects

- Blocks (Implemented as rings of links - node to node or milepost/offset to milepost/offset)

  A block is a GT-polygon defined by a ring of links in the road network.

- Areas (Implemented as rings of links and strings - node to node or milepost/offset to milepost/offset)

  An area is a G-Polygon or GT-Polygon defined by a ring containing at least one bounding segment that is not part of the road network. These boundary segments may be defined as topological links between offset points and nodes or other offset points.

C. IVHS Application Requirements

This section will review the list of IVHS applications from the U.S. Department of Transportation Program Plan for the Intelligent Vehicle Highway Systems Program (July 1993). Each application will be described by an application definition (IVHS Architecture Mission Definition, October, 1993) and will contain analysis of the navigation task requirements detailed in the previous NCGIA/Caltrans Progress Report (NCGIA, 1993). These were:

- Positioning (Where are you?)

- Route Finding (How are you going there?)

- Route Guidance (How do you get there?)

- Route Information (What's on the way?)

- Route Conditions (What's in the way?)

- Route Scheduling (When do you get there?)
- Route Dispatch (Who can get there the fastest?)
- Modal Choice (What are you going to use to get there?)

For each application there will also be an analysis of the application's spatial data level requirements. For each application there will be a description of the minimum level of spatial detail required for implementation and further levels of spatial detail required for more flexible or more advanced implementations. Some applications on the list do not require any ties to a navigable database.

1. Pre-Trip Travel Information (transit, driver, and ridesharing)

DOT System Definition

The system shall provide travelers with accurate pre-trip travel information for a complete range of multimodal transportation and transit route options upon request. The system shall provide the means to collect, manage, calculate, communicate, and present pre-trip travel information to requesting users.

Timely pre-trip information shall consist of (at a minimum) transit routes, schedules, transfers, fares, and ride-matching information. In addition, updated traffic and highway conditions shall be provided for the requested route(s). Updated (real-time) information shall consist of incidents and advisories (i.e., accidents, stalls, fires, etc.), road closures, road construction, alternative routes, traffic speeds along specific routes, parking conditions, event schedules, and weather and road conditions.

The system shall interface with traffic, route guidance, electronic payment, and transit management and information sources to obtain the necessary travel information; and provide the means and form to present the information to the requesting user. The system shall accept user information and trip parameters/preferences to calculate, communicate, and present the requested information. The information shall include modal and route preferences, trip origin and destination, departure time, arrival time, total travel time, weather and roadway conditions; and if needed, maximum number of mode transfers and intermediate stops.

Navigation Tasks

The basic function of this application is to provide comprehensive pre-trip travel information to the public. The system is designed as a decision support system to help the user decide how to get from a chosen origin to a chosen destination within a certain period of time. To support the decision making process, the system transforms the user's origin, waypoints, and destination into a routing system (positioning). The routing system generates possible transportation route alternatives (route finding) based upon the user's time constraints (route scheduling) and current road network constraints (route conditions). The system will give the user a choice of feasible automobile and public transit route alternatives (modal choice). The system will also allow access to a general information database (route information) for referencing amenities on a prospective route or to find suitable waypoints. Once the user chooses a route, the system will generate travel instructions (route guidance) for use on the trip.

Spatial Detail Needs

The level of spatial detail required by a Pre-Trip Travel Information System will consist of zero-dimensional and one-dimensional objects. Two-dimensional objects are an option. The major task of this application is route findin-. A very basic system for route finding requires a navigable database containing links and intersections for shortest path analysis of the network. On this basic network implementation, road conditions and route information could be tied directly to the links and intersection points. To perform the route guidance functions, it would be necessary for the navigable database to be able to tie information to routes. More elaborate systems would allow more precise positioning and labeling with flexible point and polygon data element types (link points, offset points, polygon points, blocks, and areas). Lanes would not be required to implement this system.

2. En Route Driver Advisory (driver information and in-vehicle signing)

DOT System Definition

The system shall provide accurate/updated traffic and highway advisory information to roadway users while enroute to their destinations. The system shall provide the means to collect, manage, and disseminate updated advisory information to roadway users through roadside and in-vehicle means. The information shall include incidents, roadway congestion, alternative routing, construction and work zones, and other hazardous environmental conditions. The system includes Driver Information and In-Vehicle Signing sub services.

The system shall interface with traffic and travel management and information sources to collect accurate roadway advisory information, and provide the administrative interface(s) to manage the information. The system shall present advisory information to roadway users through visual and audio media (roadside and in-vehicle means) consistent with safety guidelines. The system shall provide interactive user interfaces to accept requests for advisory information and to provide the media for presentation.

The system shall accept interactive user commands and information parameters which identify types of en-route travel information requested, or provide the advisory information as the information is made available through activated in-vehicle or roadway facilities.

Navigation Tasks

The basic function of this application is to provide information on road conditions (route conditions) to the driver while enroute to a destination. A secondary task is to develop alternative routes (routefinding, route guidance) around any hazard or obstruction in the current route from the current location (positioning) of the driver.

Spatial Detail Needs

The level of spatial detail required by En Route Driver Advisory Systems will consist of zero-dimensional and one-dimensional objects. The major task of this application is route conditions. On a basic level, the navigable database will tie hazards and incidents to links and intersections. Links and intersections are also required for finding alternate routes around adverse road conditions. A more advanced implementation of this system would be able to match the hazards and incidents to lanes. Another implementation option would require route level data access to analyze the route for possible road condition impacts. Polygons and non-intersection point locations would not be required to implement this system.

3. En Route Travel Advisory

DOT System Definition

The system shall provide accurate/updated transit and multi-modal information to transit users while enroute to their destinations. The system shall provide the means to collect, manage, and interactively disseminate updated scheduled and actual transit information to users through transfer/wayside points and in-vehicle/intransit means. This information shall include transit schedules and actual status, roadway and traffic conditions, and ride-matching information.

The system shall interface with transit, traffic, and electronic payment, and ride-matching/rideshare management and information sources to collect accurate transit information, and provide the administrative interface(s) to manage the information. The system shall present transit information through interactive visual and audio media. The system shall accept interactive user commands and information parameters that identify the types of transit information requested.

Navigation Tasks

The basic function of this application is to provide information to transit riders to increase the transit system’s utility to the riders. The system is designed as a dynamic user support system to help the user get to their destination in an optimal manner given the effects of traffic and transit system variables. The system must be able to locate the user in transit or at a wayside (positioning). As the conditions of the transit network change and/or the user’s location changes, the system can generate transit route alternatives (routefinding) based upon the current transit system schedule (route scheduling). The system will include a range of public transit and ride matching options (modal choice).

Spatial Detail Needs

The level of spatial detail required by En Route Travel Advisory Systems will consist of zero-dimensional and one-dimensional objects. The major task of this application is route schedule tracking based on personalized rider route needs. On a basic level, the navigable database will tie the transit route schedules to routes. For more flexible implementations, routes would be bound to links and intersections for alternative transit routing and pedestrian links. Offset points will be required to locate bus stops, transfer points, and layover points. These features could also be bound to link points. Polygons, lanes, and polygon points would not be required to implement this system.

4. Traveler Services Information (yellow pages, etc.)

DOT System Description

The system shall provide motorists, travelers, or other users with accurate information for a complete range of travel-related services and facilities upon request while in pre-trip and en-route travel modes. The system shall provide an interactive means to collect, manage, communicate, and present up-to-the-minute information on service types, locations, conditions, and status/availability to requesting users in a timely fashion. Timely traveler information shall consist of (at a minimum) identification, location, conditions, and status of food, lodging, tourist, transit, vehicle service/repair, parking, medical, police, and other "yellow pages" service facilities.

The system shall interface with traffic and transit management and information sources, as well as travel/tourist information and financial (electronic payment) services networks to collect user-requested travel information. The system shall present traveler information to users through interactive visual and audio media regardless of travel mode, i.e., in-vehicle, in-transit, at home, office, and public/private facilities. The system shall accept interactive user commands and information parameters that identify the types of traveler information and services requested.

Navigation Tasks

The basic function of this application is to provide information on roadside amenities and travel related services (route information). The information can be selected by entering the current location and route of the traveler and/or the location of likely waypoints (positioning).

Spatial Detail Needs

The level of spatial detail required by a Traveler Services Information System will require zero-dimensional and one-dimensional objects. Two-dimensional objects are an option. The major task of this application is route information. The basic level navigable database will bind travel related services and facilities to links and offset points. Lane intersections are an option to offset points. More advanced implementations could bind services to polygons (blocks and areas). Lanes would not be required to implement this system.

5. Route Guidance (includes general service plus commercial vehicle and HAZMAT specific guidance; doesn't include emergency vehicle specific guidance)

DOT System Description

The system shall provide motorists, travelers, or other users with accurate route guidance information for given user-defined travel parameters. The system shall provide an interactive means to collect, manage, communicate, and present routing information to requesting users in static (routing) and real-time (dynamic traffic and transit conditions) modes in a timely fashion. Routing information shall be representative of the requested area and consist of a presentation form which is consistent with travel safety guidelines (i.e., simple videotex, audiotex, etc.). The system shall include General Service, Commercial Vehicle, and HAZMAT service support, but not Emergency Vehicle Service specific support.

The system shall interface with traffic, transit, electronic payment, travel demand, and commercial vehicle fleet management and information sources to provide static routing and real-time traffic and transit routing information to requesting users; and provide travel demand information to the management systems. The system shall present routing information to users through interactive visual and audio means while invehicle, in-transit, or equipped facilities. The system shall accept interactive user commands and information parameters which include origin, destination, travel time, and preferred routes.

Navigation Tasks

The basic task of this application is route guidance. The system is designed to be a navigation support aid to help the user get from a chosen origin to a chosen destination. To provide this aid, the system transforms the user's origins, waypoints, and destination or the user's current location en route into a routing system (positioning). The routing system generates possible transportation route alternatives (routefinding) based upon the user's time constraints (route scheduling) and current road network constraints (route conditions). The system will give the user a choice of feasible automobile and public transit route alternatives (modal choice). These tasks are very similar to those of the Pre-Trip Travel Information System except for the en route functions.

Spatial Detail Needs

The level of spatial detail required by a Pre-Trip Travel Information System will consist of zero-dimensional and one-dimensional objects. The major task of this application is route guidance. A very basic system for route guidance requires a navigable database containing links and intersections for driving instruction generation. The links and intersection should be able to be bound into routes for future route guidance/driver advisory functions. More flexible systems would be able to access landmarks bound to offset points or link points. Very advanced systems would be able to generate driving instructions using lane level detail. Polygons would not be required to implement this system.

6. Ride Matching and Reservation (car/van pool, HOV control, etc.) DOT System Description

The system shall provide motorists with accurate ride matching or rideshare, information for given userdefined rideshare information. The system shall provide an interactive means to coordinate and administer the collection, correlation, management, and communication of rideshare information to requesting users for travel by bus, rail, carpools, vanpools, express bus, or specialized transit services. User-defined information shall include date, time, origin, and destination, and special requirements (i.e. disabled services).

The system shall interface with transit, transportation, and financial (electronic payment) management and information sources to provide ride matching or ride-share information to requesting users, and facilitate fare/credit payment services. The system shall present ride matching or rideshare information to users through interactive video or audio means. The system shall accept interactive user commands and information parameters which include date and time of travel, origin and destination, fare payment/credit, and any other special requirements.

Navigation Tasks

The basic function of this application is to match riders with transit (modal choice). The system is designed as a dynamic user support system to help the user get to their destination in an optimal manner given available modes of transit. It will be able to take user input of origin, destination, and waypoints (positioning) and departure and arrival time. The system can generate route alternatives (routefinding) based upon the current transit system schedule (route scheduling) and available fleet ready for pickup (route dispatch). Vehicles dispatched by the system (dial-a-ride, van pools, etc.) will receive pickup instructions through the system (route guidance).

Spatial Detail Needs

The level of spatial detail required by a Ride Matching and Reservation System will consist of zerodimensional and one-dimensional objects. The major task of this application is transit scheduling and dispatching based on personalized rider transportation needs. On a basic level, the navigable database will tie the transit route schedules to routes. Route finding and guidance for vehicle dispatch and passenger pickup will require links and intersections. More flexible systems will allow pickup points and landmark location bound to offset points or link points. Polygons, lanes, and polygon points would not be required to implement this system.

7. Incident Management (excludes emergency vehicle management service)

DOT System Description

The system shall collect, monitor, and evaluate traffic flow information and environmental conditions on a real-time basis to detect and correlate possible roadway incidents for coordination and incident response by appropriate agencies. Incidents shall include planned and unplanned events which affect traffic flow. The system shall facilitate and integrate traffic management operator actions (and incident management plans) for logical decision making (artificial intelligence/expert systems) and coordination of incident verification, response, removal, traffic management, information dissemination, and administrative activities with other users and agencies. This system will not include Emergency Vehicle Management support services.

The system shall interface with vehicles traveling on monitored roadways to collect traffic flow information (i.e., occupancy, queue, speed volume, etc.) and environmental conditions, and to disseminate incident (traffic advisories, alternative routes, etc.) information to motorists traveling on affected roadways. The system shall interface with traffic (and operations personnel), transit, route guidance, commercial fleet, public transportation, emergency vehicle, and traveler information management and information sources to coordinate incident information and traffic status. The system shall accept interactive traffic management operator commands and data input to coordinate incident management functions and information, and to perform system administrative activities.

Navigation Tasks

The basic task of this application is route conditions. The system is designed to support traffic management systems by tracking the location (positioning) of incidents on the transportation network. The system will interface with traffic management. and information systems to reroute traffic (route finding and route guidance) to reduce incident induced traffic delays (route scheduling). Another system function will direct non-emergency support services (route dispatch). Spatial Detail Needs

The level of spatial detail required by Incident Management Systems will consist of zero-dimensional and one-dimensional objects. The major task of this application is route condition tracking. A very basic system for incident management requires a navigable database containing links and intersections for incident location and route alternative generation. More advanced systems would be able to locate incident information on link points. To tie in off road incidents such as fires and toxic spills, offset points, polygon points, and areas would be required. Very advanced systems would locate incidents on the lane level.

8. Travel Demand Management (regulatory, mode change, parking control, emissions detection, etc.)

DOT System Description

The system shall provide roadway management control (demand prediction and high occupancy vehicle (HOV) lane use), parking management and control, air pollution/emissions monitoring, mode change support services, and travel/public awareness information services. Roadway management and control shall facilitate demand prediction, HOV priority, and electronic fare/toll collection. Parking management and control shall facilitate demand, HOV parking priority, and electronic parking fare/toll collection. Air pollution/emission’s monitoring shall facilitate collection, identification, and emission enforcement through real-time emissions analysis and route guidance on less constricted roadways. Mode change support services shall facilitate coordination of traveler ridesharing and multimodal transit itinerary options. Travel/public awareness services shall facilitate the dissemination of traffic and transportation conditions and information to better inform travelers of route and travel mode options/decisions.

The system shall interface with traffic management, electronic payment, ride matching and reservation, and pre-trip travel information management and service sources to facilitate fare/toll collection, policy and regulation enforcement, and to disseminate travel information to promote efficient roadway use. The system shall accept interactive user commands and parameters/preferences to optimized travel modes.

Navigation Tasks

The basic task of this application is reducing traffic encouraging HOV and transit ridership. While many of the strategies of Traffic Demand Management (TDM) do not require the use of a navigable database, there are a some TDM applications that do so. Once such strategies mode change support services to encourage carpooling or public transit services (modal choice). Another strategy is to route traffic around congested areas (positioning, routefinding, route guidance, and route conditions). Other TDM applications that could be tied to the navigable database include HOV lane and parking management.

Spatial Detail Needs

The level of spatial detail required by Travel Demand Management Systems will consist of zero-dimensional and one-dimensional objects. Two-dimensional objects are an option. The major task of this application is automobile trip reduction. A very basic system for generating alternate routes around congested areas requires a navigable database containing links and intersections for routing and driving instruction creation. Very advanced systems would be able to use generate routes and track HOV traffic on the lane level. Polygons (blocks and areas) could be used to store parking demand and management information.

9. Traffic Control ((includes transit priority and HOV priority)

DOT System Description

The system shall perform accurate, real-time traffic surveillance, communications, data processing, adaptive control, flow predication, and performance evaluation functions to effectively manage and optimize traffic flow throughput and efficiency. The system shall integrate and facilitate management and control of traffic flow on arterial signal and freeway/tollroad networks. The system shall utilize real-time, traffic adaptive signal control (RTTASQ systems to facilitate areawide traffic flow management. The system shall include Transit and HOV Priority services.

The system shall interface with other traffic, transit, travel demand external agencies (law enforcement, fire, medical, hazmat, etc.), incident, and information management and information sources to collect and coordinate traffic flow information and demands. The system shall interface with traffic management operators to accept commands and data inputs and present traffic and system status and performance information. The system shall interface with roadway motorists and other users for dissemination of pertinent traffic information (i.e., advisories, construction/work zones, incidents, alternate routing, etc.). The system shall accept traffic management operator commands and data inputs for traffic control strategy configuration, traffic network monitoring, and system diagnostics and administration.

Navigation Tasks

The basic task of this application is maximizing traffic flow while encouraging HOV and transit use. The Traffic Control System is a central support system for many other IVHS applications through traffic surveillance and traffic control (signal control, ramp metering, signing). While Traffic Control does not use any navigation tasks directly, the surveillance modes of traffic control (traffic detecting, aerial surveillance, police reports, etc.) collect information which must be spatially bound to the navigable database for other applications (positioning).

Spatial Detail Needs

The level of spatial detail required by Traffic Control Systems will consist of zero-dimensional and onedimensional objects. The major task of this application is adding and processing traffic network information. The lowest level of detail required for this application is links and intersections. More advanced systems will bind information to the lane level. High location precision could be achieved using link points. Polygons and routes are not required for this application.

10. Electronic Payment Services (parking, transit fares, toll collection, congestion and highway pricing, etc.)

DOT System Description

The system shall provide electronic service payment and toll collection, electronic intermodal transit fare collection, and electronic parking payment mechanisms to facilitate accurate, convenient, and efficient revenue collection. The system shall facilitate roadway pricing strategies through automated vehicle identification and electronic toll transactions. The system shall facilitate integrated electronic payment services for roadway toll collection, intermodal transit fare collection, parking payment collection, and roadway pricing.

The system shall interface with traffic, transit, travel demand, traveler service, and commercial vehicle administration management and information sources to facilitate electronic financial transactions across service providers and agencies. The system shall interface with motorists, travelers, commercial operators, and other roadway users through intermodal, electronic payment media.. The system shall accept electron user payments and information (113s) to process, route, and credit user accounts.

Navigation Tasks

The Electronic Payment Services System requires no navigation tasks. The information management needs of the system does require collection of information on toll collection, parking fees, and transit fares tied to location (positioning).

Spatial Detail Needs

The level of spatial detail required by Electronic Payment Services will consist of zero-dimensional, onedimensional, and two dimensional objects. The major task of this application is adding and processing user payment data. The lowest level of detail required for this application is links and link points or intersections for toll collections, offset points for transit fare collection. Parking fee collection data could be tied to polygon points or to blocks or areas representing the parking lots or structures. Advanced systems could record road toll data on the lane level (toll booths). Routes are not required for this application.

11. Commercial Vehicle Preclearance (includes roadside access to carrier, vehicle and driver records)

DOT System Description

The system shall provide automated regulatory and safety inspection services to preclear commercial vehicles at designated checkpoints. Automated inspection services shall collect, process, and audit vehicle and driver-specific information for compliance with carrier permits. This information shall consist of historical carrier and vehicle data, safety, operating credentials, weight, and vehicle-related history, diagnostics , and sensor data.

The system shall interface with electronic payment systems, commercial vehicle administration, fleet management and safety systems (inspection and monitoring), and vehicle collision avoidance systems (longitudinal and lateral). The system shall interface with appropriately-equipped commercial vehicles to perform weight, credential, and safety inspections. The system shall interface with inspection system operators to accept input data and commands to perform administrative regulatory and safety transactions. The system shall accept inspection system operator inputs, commands, and queries to access and audit inspection data collection and reporting.

Navigation Tasks

The Commercial Vehicle Preclearance System requires no navigation tasks. The information management needs of the system do require collection of information on vehicle inspection and fee collection systems tied to location (positioning).

Spatial Detail Needs

The level of spatial detail required by Commercial Vehicle Preclearance Systems will consist of zerodimensional, one-dimensional, and two dimensional objects. The major task of this application is adding and processing commercial vehicle inspection data. The lowest level of detail required for this application is links and offset points for locating inspection stations. Inspection stations could also be tied to polygon points or to blocks or areas representing the station sites. Routes, lanes, and intersections are not required for this application.

12. Automated Roadside Safety Inspections (automated inspection facilities)

DOT System Description

The system shall perform real-time, automated roadside safety inspections of appropriately-equipped commercial vehicles to access, collect, and audit the carrier's, vehicle's, and driver's historical safety records, rapidly and accurately inspect brake performance, and access collected and stored vehicle sensor and diagnostic data for audit at automated roadside inspection facilities. Vehicle data for storage, updates, and maintenance shall include Office of Motor Carriers (OMC) carrier ratings (including "Gold Decal" status), vehicle/driver inspection and maintenance data (including last date of inspection), verification of repairs and out-of-service records, and driver status (including licensing and citations).

The system shall interface with commercial vehicle administrative management, preclearance management, commercial vehicle fleet management and information sources, and on-board safety monitoring systems. The system shall interface with roadside inspection operators and communication systems for data transactions and regulation enforcement. The system shall accept real-time, interactive inspection operator inputs and commands to access, collect, and audit carrier, vehicle, and driver information.

Navigation Tasks

The Automated Roadside Safety Inspections System requires no navigation tasks. The information management needs of the system do require collection of information on vehicle inspection and vehicle records tied to location (positioning).

Spatial Detail Needs

The level of spatial detail required by Automated Roadside Safety Inspections Systems will consist of zerodimensional, one-dimensional, and two dimensional objects. The major task of this application is adding and processing commercial vehicle inspection data and records. The lowest level of detail required for this application is links and offset points for locating inspection stations. Inspection stations could also be tied to polygon points or to blocks or areas representing the station sites. Routes, lanes, and intersections are not required for this application.

13. Commercial Vehicle Administration Processes (electronic purchase of credentials, automated mileage and fuel reporting and auditing, and international border preclearance)

DOT System Description

The system shall provide automated commercial vehicle administration transaction processing to electronically purchase carrier credentials (annual and temporary), automatically record and audit vehicle mileage and fuel use, and facilitate automatic commercial vehicle preclearance at international border crossings. The system shall include Electronic Purchase of Credentials, Automated Mileage and Fuel Reporting and Auditing, and International Border Preclearance sub-services.

The system shall interface with electronic payment, commercial vehicle preclearance, automatic roadside safety inspection, on-board safety monitoring, and commercial fleet management and information sources to perform commercial vehicle administration activities. The system shall interface with administrative operators to provide requested information and execute clearance purchases and payment transactions. The system shall accept interactive user commands to execute selected transactions.

Navigation Tasks

The Commercial Vehicle Administration Processes System requires no navigation tasks. The information management needs of the system do require collection of information on vehicle records tied to location (positioning).

Spatial Detail Needs

The level of spatial detail required by Commercial Vehicle Administration Processes Systems will consist of zero-dimensional, one-dimensional, and two dimensional objects. The major task of this application is adding and processing commercial vehicle records. The lowest level of detail required for this application is links and offset points for locating inspection stations. Inspection stations could also be tied to polygon points or to blocks or areas representing the station sites. Routes, lanes, and intersections are not required for this application.

14. On-Board Safety Monitoring (includes driver, vehicle, and cargo)

DOT System Description

The system shall provide vehicle-based safety monitoring, diagnostic, and warning information to determine the status of the vehicle, cargo, or driver while in transit. The system shall collect, identify, and store realtime driver and vehicle status for pre- and post trip inspections. The system shall resist tampering attempts to modify stored safety information. The system shall monitor the condition of critical vehicle components (i.e., brakes, tires, engine, and lights), collision avoidance sensors, and cargo or load status for unsafe conditions (i.e., load shifting while the vehicle is in transit) against pre-determined warning thresholds; and implement safety countermeasures when thresholds are reached or exceeded (i.e., driver warning indicators). The system shall monitor driver alertness and driving time against predetermined thresholds and provide warning indicators to the driver, carrier, and enforcement agencies.

The system shall interface with critical vehicle components (tires, brakes, engine, lights, etc.), the driver, and roadside commercial vehicle preclearance, inspection facilities, fleet management, and administration information sources for information exchange and warning indicators. The system shall interface with authorized operators to interactively access, collect, and exchange safety information and warning indications. The system shall accept interactive, authorized user commands to access on-board safety information.

Navigation Tasks

The On-Board Safety Monitoring System requires no navigation tasks. The information management needs of the system do require collection of data on vehicle systems tied to location (positioning). The location of the vehicle is determined by an on board navigation system or GPS.

Spatial Detail Needs

The level of spatial detail required by On-Board Safety Monitoring Systems will consist of zero-dimensional and one-dimensional objects. The major task of this application is adding and processing commercial vehicle data. The lowest level of detail required for this application is links and intersections. The information could also be tied to the vehicle route. Polygons, lanes, and offset points are not required for this application.

15. Commercial Fleet Management (includes motor carrier and intermodal terminal operations)

DOT System Description

The system shall provide automated vehicle/mode identification, location, vehicle route guidance, and fleet administration and real-time status to facilitate fleet operations through a central dispatch/management center.

The system shall interface with traffic, transit, route guidance, and commercial vehicle preclearance, safety monitoring (roadside and on-board), and administration management and information sources to facilitate management and control of vehicle fleets. The system shall interactively interface with fleet operators to provide vehicle/mode/cargo identification, location, vehicle route guidance, and fleet administration and status. The system shall accept interactive user (dispatchers and drivers) commands and data inputs to provide real-time information and status of fleet vehicles, transit modes, and cargo.

Navigation Tasks

The basic task of this application is fleet management. The system is designed to support fleet management by tracking the location (positioning) of vehicles in the fleet. The system will assign vehicles to routes based upon demand, distance, schedules, and road network conditions (route dispatch, routefinding, route guidance, route conditions, and route scheduling).

Spatial Detail Needs

The level of spatial detail required by Commercial Vehicle Management Systems will consist of zerodimensional and one-dimensional objects. The major task of this application is fleet management. A very basic system for incident management requires a navigable database containing links and intersections for vehicle routing. Links and intersections must be bound into routes for route level data storage. More advanced systems could tie in route related information such as depots, truck stops, layovers, etc., using offset points, polygon points, blocks and areas. Lanes would not be required for this application.

16. Public Transportation Management (operation of facilities and vehicles, planning and scheduling services, personnel management)

DOT System Description

The system shall facilitate real-time intermodal, public transportation vehicle and facility operations and management, service planning and scheduling, and personnel management for fixed route transportation services through a fleet operations management center. The system shall compare real-time automated vehicle identification, location, and fleet administration status to schedule information to identify schedule variances, and calculate routing, speed, and traffic priority (signal preemption) modifications to optimize schedule adherence. The system shall collect, store, and analyze fleet performance to automatically calculate transportation system schedules to improve service routing and scheduling. The system shall automatically assign and manage personnel (drivers') parameters and preferences to routes, runs, and individual vehicles.

The system shall interface with traffic, transit (public transportation), incident, en-route transit advisory, traveler information (pre-trip, ride matching), electronic payment, safety monitoring (roadside/onboard), and travel security management and information sources to coordinate fleet management activities. The system shall interactively interface with fleet management operators to accept commands and inputs for fleet operations. The system shall accept interactive fleet operator commands and data inputs command and control operations and functions.

Navigation Tasks

The basic function of this application is to manage public transportation operations, enhancing riders' transit alternatives (modal choice). The system is designed as a dynamic transit systems support system, interfacing with routing and scheduling systems, dispatching systems, and traffic control systems. The system will be able to track vehicle location (positioning) and vehicle departure and arrival time. The system can generate route alternatives (routefinding) based upon the current traffic network (road conditions), change transit system schedules (route scheduling) and dispatch available vehicles for pickup (route dispatch). In service route changes and vehicles dispatched by the system (dial-a-ride, van pools, etc.) will receive pickup instructions through the system (route guidance).

Spatial Detail Needs

The level of spatial detail required by a Public Transportation Management System will consist of zerodimensional and one-dimensional objects. The major task of this application is transit scheduling and dispatching based on personalized rider transportation needs. On a basic level, the navigable database will tie the transit route schedules to routes. Route finding and guidance for route changing and vehicle dispatch and passenger pickup will require links and intersections. More flexible systems will allow pickup points and landmark locations bound to offset points or link points. Polygons, polygon points, and offset points would allow location of fleet yards, layover lots, service centers, and other data collection points. Advanced systems would use lane level route guidance for directions to HOV lanes.

17. Personalized Public Transit (para-transit**,** route deviations, etc.)

DOT System Description

The system shall facilitate real-time, personalized public transportation operations and fleet management by automatically monitoring fleet (vehicle and driver identification) location and current status. The system shall collect customer requests, collate ride matching, vehicle run routing, and customer feedback; and provide real-time fleet operations communications for route modification instructions and schedules. The system shall accept, process, analyze, and store multiple customer requests (and electronic payment) for personalized shared transportation from specified origins to specified destinations within given time frames -The system shall automatically correlate and optimize customers' origin-destination (O-D) pairs, times, and special accommodations (e.g., wheelchair access) into vehicle driver instructions (Le, customers, shared vehicle ID's driver route guidance, and schedule information) for both real-time and pre-planned operating modes. The system shall automatically assign and manage personnel (drivers') parameters and preferences to route areas, runs, and individual vehicles.

The system shall interface with traffic, transit, traveler information (pre-trip, en-route advisory), ride matching and reservation, electronic payment, and public travel security management and information sources to collect, correlate, and process user travel requests. The system shall interface with customers to accept interactive user requests and information (O-D, departure/arrival, special requirements, etc.); and with fleet operators and drivers for commands and data. The system shall accept interactive fleet operator commands and data inputs for command and control operations and functions.

Navigation Tasks

The basic function of this application is to manage custom routed public transportation operations, enhancing riders' transit alternatives (modal choice). The system is designed as a custom transit dispatching support system, interfacing with routing and scheduling systems. The system will be able to track vehicle location (positioning) and vehicle departure and arrival time. The system can generate route alternatives (route finding) based upon the current traffic network (road conditions), schedule departures (route scheduling), and dispatch available vehicles for pickup (route dispatch). Dispatched vehicles will receive pickup instructions through the system (route guidance).

Spatial Detail Needs

The level of spatial detail required by a Personalized Public Transit System will consist of zero-dimensional and one-dimensional objects. The major task of this application is transit dispatching based on personalized rider transportation needs. On a basic level, the navigable database will tie personal transit needs to routes. Route finding and guidance for vehicle dispatching and passenger pickup will require links and intersections. More flexible systems will allow pickup points and landmark locations bound to offset points or link points. Polygons, polygon points, and offset points would allow location of fleet yards, layover lots, service centers, and other data collection points. Advanced systems would use lane level route guidance for directions to HOV lanes.

18. Emergency Notification and Personal Security (driver and personal security, automated collision notification, HAZMAT incident notification)

DOT System Description

The system shall provide efficient and accurate notification of incidents and other emergency situations to appropriate response authorities. The system shall receive emergency notification from properly-equipped vehicles (or personal devices) to determine the time and location of the incident/emergency, the nature and status conditions, and the characteristics of the vehicle or cargo; formulate proper response plans; and notify appropriate responding agencies to manage the incident/response on-site. The system shall provide driver and personal distress notification and acknowledgement for assistance and non-injury response coordination for minor property damage situations and mechanical breakdowns (failure, flat tire, no gas, etc.). The system shall provide for incident location, nature, and notification/cancellation. The system shall provide automatic notification and response for serious injury-related incidents or emergencies and forward vehicle location, nature, and conditions/status. The system shall provide accurate hazardous material incident notification, response determination, and coordination to appropriate enforcement agencies and HAZMAT teams. The system shall provide incident information such as location, nature and hazardous material type. The system shall include Driver and Personal Security, Automated Collision Notification, and HAZMAT Notification sub-services.

The system shall interface with traffic, incident management, route guidance, commercial fleet, public transportation, electronic payment, and emergency vehicle management and information sources, and on-board vehicle sensor systems to communicate and coordinate information between responding agencies. The system shall interface with notification users (where manually controlled) to activate signal transmission. The system shall accept user commands to activate (where manually controlled) notification transmissions.

Navigation Tasks

The basic tasks of this application are incident location (route conditions) and emergency dispatching (route dispatch). The system is designed to support emergency notification and personal security systems by reporting the location (positioning) of incidents on the transportation network. The system will interface with the Traffic Management and Information System to reroute traffic (routefinding and route guidance) to reduce incident induced traffic delays (route scheduling). Spatial Detail Needs The level of spatial detail required by Emergency Notification and Personal Security Systems will consist of zero-dimensional and one-dimensional objects. Two-dimensional objects will be an option. The major task of this application is route condition tracking. A very basic system for incident management requires a navigable database containing links and intersections for incident location and route alternative generation. More advanced systems would be able to locate incident information on link points. To tie in off road incidents such as fires and toxic spills, offset points, polygon points, and areas would be required. Very advanced systems would locate incidents on the lane level.

19. Public Travel Security

DOT System Description

The system shall provide automate notification, communications, and monitoring of public transit vehicles and facilities (stations, parking lots, etc.) through visual and audio sensors to notify appropriate responding agencies. The system shall accept notification form manually actuated sensors and personal communication systems/devices.

The system shall interface with en-route transit, ride matching and reservation, public transportation, and electronic payment management and information sources to receive user-actuated notification transmissions, locations, and coordinated responses to law enforcement and other appropriate agencies. The system shall accept user-actuated notification signals for assistance requests to responding agencies. Navigation Tasks The Public Travel Security System requires no navigation tasks. The system does transmit information to security systems on vehicle or incident location (positioning).

Spatial Detail Needs

The level of spatial detail required by Public Travel Security Systems consists of zero-dimensional and onedimensional objects. The major task of this application is adding and processing incident location data. The lowest level of detail required for this application is links, intersections, and offset points for locating incidents. Link points could be added for greater precision. Routes could also be included for tracking route incident levels. Polygons and lanes are not required for this application.

20. Emergency Vehicle Management (fleet management, route guidance, signal priority)

DOT System Description

The system shall provide the emergency information collection, emergency nature and location determination, response type, and route guidance and information coordination for accurate agency deployment. This system includes Emergency Vehicle Fleet Management, Route Guidance, and Signal Priority sub-services.

The system shall interface with traffic, transit, traveler information, incident management, route guidance management and information sources for coordination of notification and response.

Navigation Tasks

The basic tasks of this application are incident location (route conditions) and emergency dispatching (route dispatch). The system is designed to support emergency vehicle dispatch systems by reporting the location (positioning) of incidents on the transportation network. The system will interface with the Traffic Management and Information System to reroute traffic (route finding and route guidance), and change signaling to reduce incident induced traffic delays and speed emergency response (route scheduling).

Spatial Detail Needs

The level of spatial detail required by Emergency Vehicle Management Systems will consist of zerodimensional and one-dimensional objects. Two-dimensional objects will be aii option. The major task of this application is emergency vehicle dispatch. A very basic system for vehicle dispatching requires a navigable database containing links and intersections for incident location and route alternative generation. More advanced systems would be able to locate incident location on link points. To tie in off road incidents such as fires and toxic spills, offset points, polygon points, and areas would be required. Very advanced systems would locate incidents and route emergency vehicles on the lane level.

21. Longitudinal Collision Avoidance (rear-end crash warning and control, autonomous intelligent cruise control, cooperative intelligent cruise control, head-on crash warning and control, passing warning (on two-lane roads), backing crash warning)

DOT System Description

The system shall provide active longitudinal collision avoidance functions to prevent vehicle collisions. The system shall provide driver warning notification through visual, tactile, or audio means, provide temporal partial vehicle control, or full vehicle control to support the following sensing capabilities: Autonomous Intelligent Cruise Control, Cooperative Intelligent Cruise Control, Rear-end Crash Warning and Control, Head-on Crash Warning and Control, Passing Warning, and Backing Crash Warning.

The system shall interface with the vehicle driver and other stationary or moving objects in the pathway of the vehicle.

Navigation Tasks

This application does not require any navigational tasks or interface with a navigable database.

22. Lateral Collision Avoidance (lane change/blind spot crash warning and control, lane keeping warning and control)

DOT System Description

The system shall provide active lateral collision avoidance functions to prevent vehicle collisions. The system shall provide driver warning notification through visual, tactile, or audio means, provide temporal partial vehicle control, or full vehicle control to support the following sensing capabilities: Lane Change/Blind Spot Situation Display Crash Warning and Control and Lane Keeping Warning and Control.

The system shall interface with the vehicle driver and other stationary or moving objects located in lateral regions around the vehicle.

Navigation Tasks

This application does not require any navigational tasks or interface with a navigable database.

23. Intersection Crash Warning and Control

DOT System Description

The system shall provide active intersection collision avoidance functions to prevent vehicle collisions. The system shall provide driver warning notification through visual, tactile, or audio means and provide full vehicle control to support the intersection collision avoidance. The system shall interface with traffic intersection sensing functions and the vehicle driver for avoidance responses.

Navigation Tasks

This application does not require any navigational tasks or interface with a navigable database.

24. Vision Enhancement for Crash Avoidance (inclement weather and at night)

DOT System Description

The system shall provide active driver vision enhancement functions to prevent vehicle collisions during low visibility conditions. The system shall provide driver warning or notification through visual, tactile, or audio means to support improved visibility during night or extreme environment conditions. The system shall interface the driver for avoidance responses.

Navigation Tasks

This application does not require any navigational tasks or interface with a navigable database.

25. Impairment Alert (impaired driver warning and control override, vehicle condition warning, in-vehicle infrastructure condition warning, integrated warning systems)

DOT System Description

The system shall provide active vehicle monitoring and roadway condition sensing to provide driver warning notification through visual, tactile, or audio means and provide temporal partial vehicle control to support the following sensing capabilities and conditions: Impaired Driver Warning and Control Override, Vehicle Condition Warning, In-Vehicle Infrastructure Condition Warning, and Integrated Warning Systems. The system shall interface the vehicle driver for corrective responses.

Navigation Tasks

This application does not require any navigational tasks or interface with a navigable database.

26. Pre-Crash Restraint Deployment

DOT System Description

The system shall provide active vehicle crash monitoring and sensing to deploy driver restraint systems.

The system shall interface with the vehicle driver for safety responses.

Navigation Tasks

This application does not require any navigational tasks or interface with a navigable database.

27. Fully Automated Vehicle Operation (Automated Highway Systems)

DOT System Description

The system shall provide full-featured automated highway guidance and control functions for both infrastructure and vehicles. The following functions shall be integrated into Automated Highway Systems:

Longitudinal Collision Avoidance, Lateral Collision Avoidance, Intersection Crash Warning and Control, Vision Enhancement, Impairment Alert, and Pre-Crash Restraint Deployment. The system shall interface with infrastructure and vehicle based subsystems and the driver.

Navigation Tasks

The basic task of this application is vehicle navigation. The system is designed to automated vehicle operation by combining extremely precise vehicle location with on board sensor systems (positioning). The system will create vehicle routes based on parameters of origin, destination, waypoints, time constraints, distance, and road network conditions (route finding, route scheduling, and route dispatch).

Spatial Detail Needs

The level of spatial detail required by Automated Highway Systems will consist of zero-dimensional and one-dimensional objects. The major task of this application is vehicle navigation on a very high level. The minimum system requirement for automated vehicle operation is a navigable database containing lanes, links and intersections for vehicle routing. Route level information is also needed for individual vehicle tracking. Polygons and offset points would not be required for navigation purposes.

Table 2: DOT Application Navigation Tasks

	Applications	Positioning	Route Fndng	Route Guidnce	Route Info	Route Cond	Route Sched	Route Dispat	Modal Choice
1	Pre-Trip Travel Info	X	X	X	X	X	X		X
2	En Route Driver Advisory	X	X	X		X			
3	En Route Travel Advisory	X	X	X	X		X		X
4	Traveler Services Info	X			X				
5	Route Guidance	X	X	X		X	X		X
6	Ride Matching and Reservation	X	X	X			X	X	X
7	Incident Management	X	X	X		X	X	X	
8	Travel Demand Management	X	X	X		X			X
9	Traffic Control	X							
10	Electronic Payment Services	X							
11	Commercial Vehicle Preclearance	X						X	
12	Auto Roadside Safety Inspections	X						X	
13	Commercial Vehicle Administration	X						X	

14	On-Board Safety Monitoring	X							
15	Commercial Fleet Management	X	X	X		X	X	X	
16	Public Transportation Management	X	X	X		X	X	X	X
17	Personalized Public Transit	X	X	X		X	X	X	X
18	Emergency Notification & Personal Security	X	X	X		X	X	X	
19	Public Travel Security	X						X	
20	Emergency Vehicle Management	X	X	X		X		X	
21	Longitudinal Collision Avoidance								
22	Lateral Collision Avoidance								
23	Intersection Crash Warning								
24	Vision Enhancement for Crash Avoidance								
25	Impairment Alert								
26	Pre-Crash Restraint Deployment								

27	Automated Highway Systems	X	X				X	X	

Table 2: DOT Application Navigation Tasks

	Applications	Inter-sectn	Link Pnts	Polygn Points	Offst Pnts	Links	Lnes	Rtes	Blcks	Areas
1	Pre-Trip Travel Info	***	xxx	xxx	***	***		***	xxx	xxx
2	En Route Driver Advisory	***				***	xxx	xxx		
3	En Route Travel Advisory	xxx	xxx		***	xxx		***		
4	Traveler Services Info	xxx			***	***				
5	Route Guidance	***	xxx		xxx	***	xxx	***		
6	Ride Matching and Reservation	***	xxx		xxx	***		***		
7	Incident Management	***	xxx	xxx	xxx	***	xxx		xxx	xxx
8	Travel Demand Management	***				***	xxx		xxx	xxx
9	Traffic Control	***	xxx			***	xxx			
10	Electronic Payment Services	xxx	***	xxx	***	***	xxx		xxx	xxx
11	Commercial Vehicle Preclearance			xxx	***	***			xxx	xxx
12	Auto Roadside Safety Inspections			xxx	***	***			xxx	xxx
13	Commercial Vehicle Administration			xxx	***	***			xxx	xxx

14	On-Board Safety Monitoring	***				***		xxx		
15	Commercial Fleet Management	***		xxx	xxx	***		***	xxx	xxx
16	Public Transportation Management	***		xxx	xxx	***	xxx	***	xxx	xxx
17	Personalized Public Transit	***		xxx	xxx	***	xxx	***	xxx	xxx
18	Emergency Notification & Personal Security	***	xxx	xxx	xxx	***	xxx		xxx	xxx
19	Public Travel Security	***	xxx		***	***		***		
20	Emergency Vehicle Management	***	xxx	xxx	xxx	***	xxx		xxx	xxx
21	Longitudinal Collision Avoidance									
22	Lateral Collision Avoidance									
23	Intersection Crash Warning									
24	Vision Enhancement for Crash Avoidance									
25	Impairment Alert									
26	Pre-Crash Restraint Deployment									

27	Automated Highway Systems	***				***	***	***		

** is required spatial data level, xx is optional spatial data level

III. PLANAR DATA MODEL

A. Introduction

This portion of the report describes the planar data model and its potential for supporting a navigable geographic database for intelligent vehicle highway systems (IVHS) and other advanced road and traffic management systems. It is an extension of one portion of the National Center for Geographic Information and Analysis (NCGIA) Progress Report for Caltrans Agreement 65TI55 (NCGIA, 1993). Its coal is to report on the results of the current NCGIA research on potential data models for IVHS and to describe the objectives for and products from the continuing prototype. It provides a detailed investigation of the planar data model's applicability to IVHS, but it omits discussion of the other potential data models discussed in the Progress Report. This section begins with introductory remarks about the nature and function of data models to provide some continuity with the Progress Report and to make this document complete for the reader independent of other documents.

B. Purpose of the Data Model

When discussing potential data models for a set of applications such as IVHS, it is important to understand how the data model functions and how it influences the performance of the applications. The data model provides the conceptual structure for the database that supports the applications. In this case, the database and data model refer to the geographic database that is the foundation for IVHS systems. The geographic database provides the data necessary for routing and other important tasks in IVHS. Therefore the database must be consistent and comprehensive, and it must be structured in a way that facilitates the highway system and traffic management tasks that it will support. It must also effectively replicate the way that the users of the database conceptualize the road network.

There are many possible ways to structure geographic data and many data models are possible. The Progress Report discussed five different data models. Some of them were derivatives of a fundamentally planar model and one was a non-planar model with intelligence about relationships between linear elements embedded within it. 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 is the planar model is the least complex and most efficient model for databases of areal features; and another is the conceptual similarity between the planar model and the familiar road or street map. As will be discussed, this model has certain advantages including its simplicity and its widespread use for certain analytical capabilities, but it has weaknesses as well. The following section is a recapitulation of general concepts of data models.

Data Model Function

A data model determines the way in which the data in a database are organized. It is "a general description of specific sets of entities and the relationships between these sets of entities" (Peuquet, 1984). The data model 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. For this report we are concerned primarily with spatial data and attribute information associated directly with them. So the data model must address how to represent spatial entities, e.g. roads, and relationships between them and how to relate nonspatial attribute information to the spatial objects. Since it involves human interpretation of the real world and is often tailored to a given application, different users and different applications may have different data models to represent the same phenomena (Peuquet, 1984).

There are several levels of data modeling, differing in their degree of abstraction, and identified as external, conceptual, logical, and internal. The external level entails a selection of the real world features and the data about them that are relevant to the problem. The conceptual level involves a formal schematic representation of how the features are related. The conceptual model is translated or mapped into the logical models which are the mathematical structures used when storing the data in a computer. The internal level deals with the byte-level data structure of the database, i.e. storage devices, file structures, access methods and locations of data in the storage devices (Laurini, 1992). The logical and internal models are sometimes referred to as data structure or file structures. This report is concerned with the external and conceptual level models. These are the levels that deal with how people view the phenomena being modeled.

Each model provides a specific set of possible elements that it can manipulate. For example, the internal level of data models is restricted to the means of storing digital data (bytes, words, sectors, etc.) specific to the hardware being used. If one considers the different levels of data models as a hierarchy with the external model at the top, 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 a smaller number of elements maintained by the lower levels of models can encode the more varied features present in the levels above (Milne et al., 1993). In the case of a planar conceptual data model for an IVHS database, the features available are different kinds of point and line elements. As described in detail in later sections, each type serves different functions and can be considered as having different behaviors. In general, the critical elements available can be classified as nodes and links. One objective of this research is to produce a structure that can use these elements to represent the large variety of road, turn, and lane configurations that exist and to do it in a way that can support efficient queries and analyses of data about them.

The data model is extremely important to the efficacy of the database because it defines which features are stored in the database, how they are stored, and how relationships between the entities are stored. In an extreme case, if a data model omits information necessary for an application, the application is not feasible. Other requirements of a data model are more subtle, however. For example, the data model must answer questions of whether roads should be stored as a series of individual segments or as one long line. In addition, the data model must consider whether data should be stored explicitly or implicitly. That is, should the database store particular data as an additional element or should it generate the data when needed for certain functions?

The way features and relationships between them are modeled affects the performance of the database as well. It determines the basic units that the database will work with and how it will manipulate those units for analytical purposes. For example, if the model represents a road as several independent segments, it is fairly easy to associate information for portions of the road **that are** conterminous with the road segments. However, the database treats these segments as independent so it is difficult to analyze the entire road length. On the other hand, if the database stores the road as one line, the entire road can be treated as a unit, but some additional structure is necessary to associate data with segments of the road. Given the impact of the data model on the ability to use the data, it is important to have a good idea of the applications that the data will support.

Clearly the data model must include all of the entities and relationships necessary for the uses which drive the development of the model. Researchers have identified several additional criteria for evaluating a data model. Critical among them are comprehensiveness, robustness, and efficiency. The model must be robust enough to handle anomalous or special circumstances, and it should be efficient in terms of data storage and processing.

C. The Planar Data Model

The data models considered for this research are based on a single line (center-line) representation of the roads. This is necessary 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. For example, a map scale of 1:24,000 implies a spatial accuracy of approximately 12 meters or 40 ft. according to the U.S. Geological Survey (Federal Geodetic Control Subcommittee, 1989). Such data are available from several sources. Representing road widths truthfully requires a minimum map scale of 1: 1200 (I inch = 100 feet), although an even larger scale would be preferable.

General Description of the Planar Model

The planar data model is derived from the concept of a planar graph. In this model, every intersection of lines must be indicated by a node. 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. While connections between links can be derived from the topological structure, the model stores the links as independent elements with no intelligence regarding relationships to other elements. This is discussed in greater detail in subsequent sections.

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 on turning from one road onto another. If two links are connected, the model allows turns from one to the other. However, there are enhancements that explicitly store turn restrictions.

This model is most applicable in urban setting for several reasons. First, in urban areas people are more inclined to consider the individual links of a road rather than the road in its entirety. Second, 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. Thus, a good example of an appropriate application of the planar model is a city street network like that of downtown Santa Barbara. In this case, all of the streets are split into separate segments terminated at nodes.

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, such as those for roads that cross Route 101 in Santa Barbara, 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. Because the data model itself does not restrict illegal or physically impossible turns, such restrictions must be encoded explicitly in an additional table.

An alternative to the planar data model is a nonplanar model. Future research will investigate the benefit of the nonplanar model, but it is useful to describe it generally in this report to contrast it with the planar data model. The most significant difference exhibited by the nonplanar model is that lines can cross without creating an intersection. Similar to the planar model, the basic elements may still be links and nodes with linear elements terminated by two nodes. However, the link-node terminology is usually not associated with this model. The nonplanarity of this model adds an implied third dimension in that if two lines cross without intersecting, the implication is that they are at different elevations, i.e. they are grade separated. The nonplanar model introduces other benefits and efficiencies as well, but they are not the subject of this report. Briefly they include more efficient geocoding, better compatibility with the general conceptual view of roads, and the capability of maintaining other transport networks, like rail, topologically separate.

While it is possible to make general comparisons at this point, it would be more informative to leave comparisons to the future work. Thus, detailed contrasts will be the subject of additional research.

Details of the Planar Model

Before discussing the actual elements associated with the planar data model, it is useful to put the discussion in the context of data modeling introduced earlier. The problem of developing a data model is one of mapping the various entities that exist in the phenomena of interest onto a more limited set of objects used by the model. In the case of this research, the problem is that there is an extremely large number of different types of roads and road combinations; the planar data model offers only nodes and links to represent this wide variety. However, it is possible to classify the road configurations into general categories and apply the objects of the planar model to these classes.

In addition to this aggregation, the planar model functions as well as it does because its elements serve several roles. The basic objects of the model can provide physical, topological or behavioral functions. Physical functions are those where the objects represent entities that physically exist in the world. The most obvious example of this is links portraying roads. Topological functions provide the logic and intelligence of the database. Topology is the storage of qualitative spatial relationships between elements, such as connectivity, adjacency, etc. Thus nodes may provide topological functions by indicating where links intersect. Behavioral functions are those that model possible driving decisions or behaviors. Although a portion of a road is not physically detached, the model may represent it independently because of a distinct driving behavior associated with it. Decisions are required about how the objects may model any of these functions but the greatest number of questions arises with the behavioral functions.

The following is an in depth consideration of each type of object in the planar data model and its functions.

Objects in the Planar Model

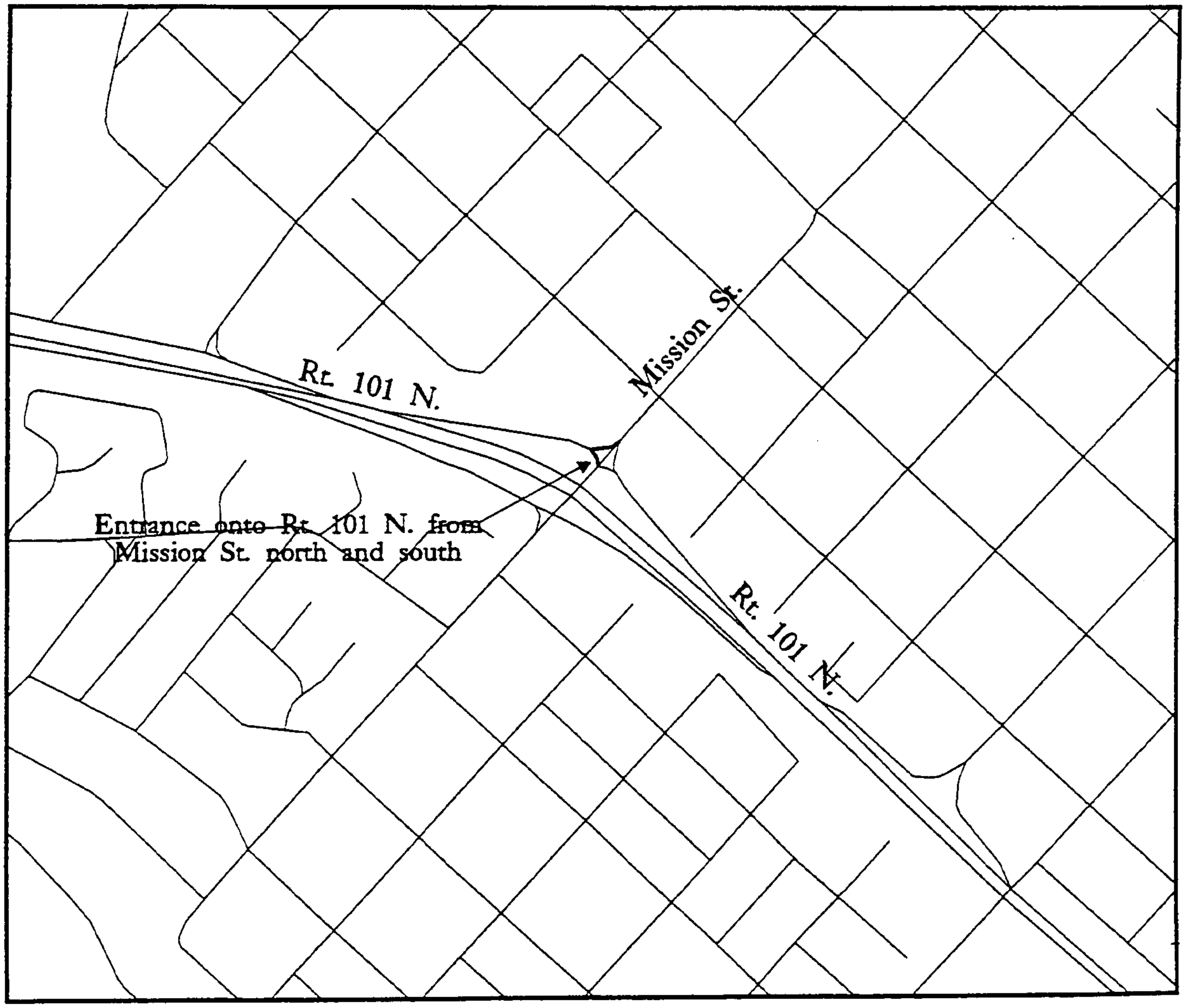
As mentioned, the planar model is constructed from links and nodes. They each have individual functions as well as prescribed relationships to each other. Links are used to represent any linear map feature, and nodes exist where links begin, end or cross (which in the planar model implies the beginning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

Links in the planar model represent road segments from one intersection to the next. The centerline implementation of the planar model that is discussed here represents the entire width of streets with a line that has no width. This is an issue when it is necessary to store information related to the road width such as lane information. 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.

Links serve obvious physical functions in that they are a linear representation of streets. They also perform behavioral functions. Links can be used to indicate turns from one link to another that may occur in separate lanes and are significant enough to warrant individual treatment and information. An example of this is Navigation Technologies, Inc.'s (NavTech's) representation of the turn from Mission Street onto the entrance ramp for Route 10 1 North (see Figure 1). Another behavioral function is where access to or egress from a road is limited in certain directions. 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. One example of this in Santa Barbara is the length of State Street that is divided with an island. The NavTech database represents this stretch with two sets of links, one for each direction of State Street (see Figure 2). In this case, the physical and behavioral functions overlap. Links may also perform topological functions. For example, street segments that restrict left turns are represented by two sets of links. A left turn option that may appear in the middle will have to be represented by a small link that connects the links for each direction of the road. Again the State Street example is illustrative. Short links connect the two directions of State Street where there is a break in the island.

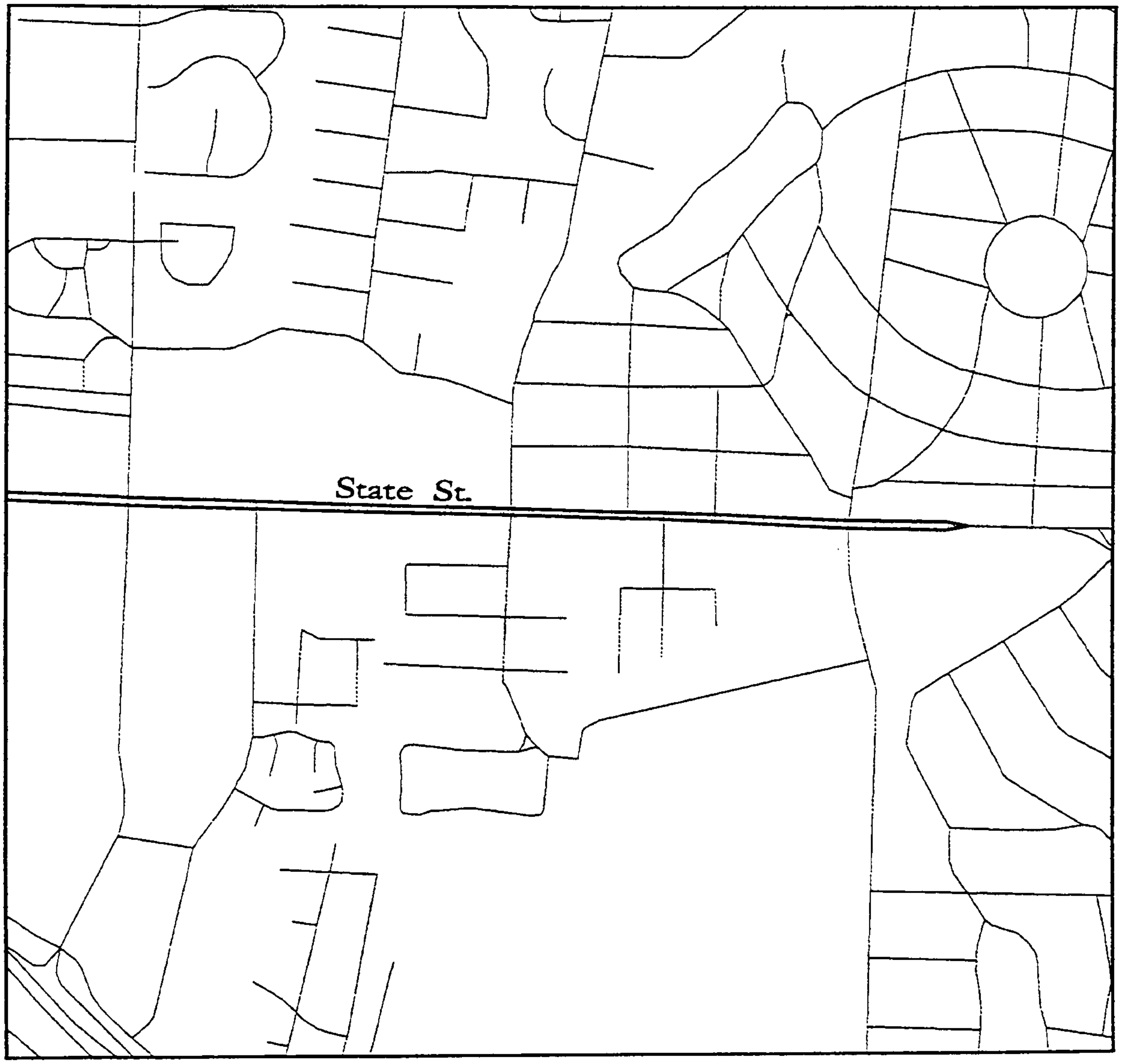
Nodes in the planar model function as terminating points for links. They represent intersections because if more than two links terminate at the same node, then they are connected and intersect. In this description, the links are separate objects so they do not intersect in the common sense. When we talk about road intersections, we are actually thinking about two continuous linear features crossing each other, creating some additional entity where they cross, but maintaining their continuity as features. The node in the planar model represents the entity of the intersection, but it also splits the linear features into discontinuous links. 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. Nodes can serve topological functions even when a physical intersection does not exist. 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
Representation of Mission St. Entrance onto Rt. 101 from Navigation Technologies Data



This portion of the NavTech data set shows the addition of links to represent driving decisions not physical entities. There are two separate links from Mission St. to the entrance ramp for Route 101 North, but there is only one physical connection between Mission St. and the entrance ramp.

Figure 2
Representation of Upper State St. from Navigation Technologies Data



This portion of NavTech's database shows the representation of upper State St. with two sets of links, one set for each direction. This simulates the physical configuration of this part of State St., and it also models the left turn restrictions from each direction of State St.

Nodes can serve a behavioral function when they represent decision points. They may be added to make obvious driver conditions explicit to the database. While the node may represent the decision point, it exists because there is potential for a turn in the network. So nodes represent decision points because there is a convergence of links on the point. It is important to note that the decision location at a node is forced to be a point which may be less forgiving than the actual extent of the possible decision. Similar to the function of links, nodes exist in the database where they do not in reality because the planar model is not a three-dimensional model.

The relationship between specific links and nodes is the basis of topological structure in the database. While the planar model does not necessarily contain topology, 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

Most GIS implementations use relational database management systems to store attribute information for spatial objects. The attribute information is often nonspatial, such as road capacity. However, it can be derived from spatial information as well, e.g. road length. In the relational model, each entity is a record or tuple. In the planar data model, the basic objects are links and nodes, so each link or node is a record in the relational table that stores the attributes. Each separate attribute is a field in the table. GIS software packages that maintain attribute information as well as spatial information typically provide a tight coupling or full integration with a database manager. This makes it possible to query and manipulate both the spatial and attribute information almost seamlessly.

While the planar model can take advantage of relational data structures to store information about entire links, it has more difficulty handling attribute information that refers only to 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 the attributes change along a link, this requires two or more instances of the attribute for the link, which translates to two records in the attribute table for the link. However, the link as the basic unit should have only one record in the table to store its information. 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.

Some Implementations of the Planar Model

Two of the best known implementations of the planar model are those used by the Arc/Info software package and the TIGER data structure developed by the United States Bureau of the Census. They are very similar in the primitive spatial elements that they use and the way that they store the relationships between them, but the data structures or logical and internal models that they use to encode their planar models differ. In addition, the Arc/Info data structure is capable of some nonplanarities. The following provides a brief description of each implementation to clarify how the conceptual planar model translates to an actual database or data structure. The Progress Report contains a detailed description of the TIGER structure, so this report should be consulted if more depth is desired.

TIGER

For the 1990 census, the Census Bureau released the Topologically Integrated Geographic Encoding and Reference (TIGER) database. This was based on the 1980 Dual Integrated Map Encoded (DIME) data, but it covered the entire contiguous United States while the DIME data included only Metropolitan Statistical Areas (MSA's). The TIGER data is also in a slightly different format. From the user point of view, however, the conceptual format is the same. The DIME/TIGER format represents all linear features (i.e. roads, streams, power lines, etc.) as single line elements. Presumably this corresponds to the center line of the feature, but it may simply be a function of the compilation scale. That is, at 1: 100,000 such features look like a single line.

Technical descriptions of the TIGER file format refer to the primitive elements as O-cells, I -cells and 2-cells. These are nodes, lines and areas respectively. The previous data model discussion has not considered polygons, but the planar structure is the most effective model for bounding and representing areas. As with the TIGER data or other urban data, it would be possible to construct polygons corresponding to city blocks from the roads that bound them. The Census Bureau considers shape points, or line vertices, a separate entity of less significance to the geographic structure. The TIGER format includes a set of rules regarding relationships between the 0-, 1- and 2-cells to ensure topological integrity. For example, a 1 -cell is always surrounded by one or two 2-cells and terminates with two 0-cells. Census descriptions of the TIGER data model have included a series of pointers between the 0-, 1-, and 2-cells, although the actual TIGER data distributed by the Census Bureau does not contain this pointer structure. In the depictions of the pointer structure 1-cells point to 0-cells that terminate them, other I-cells that are connected to the same one cells, and 2-cells that the I-cells bound (Marx, 1986).

Arc/Info

Arc/Info is actually the tight coupling of two large sets of software, Henco's (Waltham, Massachusetts) Info, a relational database management system, and Arc, the graphic editing, display, management, and analysis routines written by Environmental Systems Research Institute of Redlands, California. The Arc/Info data structure is based on a planar data model, although it is now possible to have nonplanarities in a network database. The Arc/Info system was developed primarily for polygon processing such as that required to automate the "McHargian" overlay analysis. The polygon data structure required was planar enforced meaning no polygons could overlap and the entire spatial extent of the database was covered by one polygon. This structure resulted in absolute planarity among the linear elements which originally functioned only as boundaries for the polygons. Similar to the TIGER structure, Arc/Info uses nodes and links (which it calls arcs) as its primitive elements. The links join two nodes but may contain vertices that give shape to the arc.

One of the main differences between Arc/Info and the TIGER file is their means of identifying areas. The TIGER file identifies a polygon as the set of links that bound an area of some characteristic, so the identify of polygons is embedded in the arcs that bound them. Arc/Info stores polygons as spatial entities, but these entities are identified by a label point within them. So Arc/Info considers polygons as the area that surrounds a label point of a certain characteristic out to the links that enclose it. Thus the polygons have more importance in the Arc/Info structure, and it is the label point that provides the bridge between the geographic object and its information in an attribute table. Arc/Info stores all of the coordinate information and topological relationships in a series of Info tables with pointers between them.

Functionality of the Planar Model

Different data models provide different views of the same phenomena. Because the difference between data models is a result of the models' objectives, it is important to understand the requirements and uses of the navigable database when developing and evaluating the data model to be used. The data model under consideration here must support the routing, traffic management, and automatic highway information functions included in consideration of IVHS. This section summarizes how the planar model is advantageous or disadvantageous for the main IVHS applications.

The applications that are often included within 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 procedures relating to vehicle routing. The procedures include finding the best route (route finding) from a given point to another specified point 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 finding has been a subject of research in the field of operations research and network analysis for decades. Most of the shortest path route finding algorithms apply to a planar graph. 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. Geocoding is the process of giving x,y coordinates to entities based on their location along a link or set of links. Address matching is often used synonymously because the most common application of geocoding is locating houses based on their street address. Clearly this process requires that address range information be stored for each link.

Advantages and Disadvantages of the Planar Model

One of the most significant advantages of the planar model is that it is common and fairly simple. Links and nodes may provide several different functions in the mapping from an external model to a conceptual model, but 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.

Another advantage of the planar model with respect to IVHS applications is that many of the routines that are part of the applications employ algorithms based on the planar model. For example, many shortest path algorithms have been written for planar graphs. 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 routing because of its 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. Generally, it has not been determined whether the additional overhead introduced by the planar model results in significant inefficiencies.

In addition to the processing overhead, the planar structure does not replicate how people conceptualize the road network. 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. The view of the road data may not follow the structure imposed by the links at all. For example, attributes such as road paving may change due to expediency, chance or external factors such as jurisdictional changes.

D. Issues in Using the Planar Model for Caltrans IVHS

To this point, this portion of the report has considered the characteristics of the planar data model in general with some discussion of the benefits of the planar model to certain network applications. However, the mandate of this research was to investigate the most effective data model for IVHS applications. In addition, UCSB 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 following section considers the issues that arise when including lane information in the database and enhancing the database to support the special nature of IVHS applications.

Lanes in the Planar Model

The problem posed when introducing lanes into the planar data model, or any single line model for a road network, is that the model requires a hierarchical relationship between links and lanes. There are potentially many lanes to each link. Because the accuracy required to represent the lane geometry necessitates exorbitant cost, it is unlikely that the lanes will be represented accurately. It may be possible to represent the lane topology graphically, however this makes tasks dependent on the geometry represented in the model more complex. In addition, connections between lanes at intersections can be exceedingly complicated to represent graphically. Thus, this effort will concentrate on incorporating lane relationships into a single line spatial model.

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

In attempting to determine the most effective method for encoding lanes, it is important to understand why lane information is useful. There are many ways to integrate lanes into the model, but the most appropriate way will be dependent 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. In fact Navigation Technologies, in a special application for Siemens, handled lane information in additional tables that were related to specific intersections. Turn restrictions, time of day restrictions and lane geometry can all be stored based upon link class and distance from intersection.

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. Lane monitoring is required for evaluating and controlling implementation.

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 enumerated above in the manner that most efficiently fits within the single-line representation of roads in either a planar or nonplanar data model. This report addresses only the planar model. Future research will concentrate on the nonplanar model. The database will not represent the geometry of lanes due to the cost involved, so the important lane information must be stored implicitly. The following discussion addresses possible methods of representing lane information on the road network. It will progress from the purely planar approach to more sophisticated methods. The ultimate decision about which method is most appropriate will depend on the objectives for the database with respect to both the application objectives and the logistical objectives related to computer storage, e.g. minimizing database size.

There are several complexities that relate to the objectives for including lanes in the model, but do not correspond to them on a one-to-one basis. These considerations provide the basis for the following discussion. 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 changed. 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 the number 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 is the complement of the previous method, in that this one views the lanes for each link as independent entities to be modeled 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.

The portion of the link that is covered by a certain number of lanes can be encoded two ways. The first stores the be-inning and ending points of the number of lanes as offsets from the beginning node of the link, i.e. with an attribute field in the lane database for each offset. The second stores the percentage or total length of the link covered by the number of lanes from the point where the number of lanes changes. The lane table would contain one field that records the length of the number of lanes from the point of change (see Figure 3). For example, if the number of lanes changes from two to one 250 feet along a 500 foot link, the two-lane portion would be stored with a beginning offset of 0 and an end offset of 250. The one-lane portion would have a starting offset of 2,50 and an ending offset of 500. Conversely, the two-lane portion of the link may be stored simply with one entry of 250 representing that it extends 250 feet along the link. The one-lane portion would also be given the value of 250 because it covers 250 feet of the link as well. The advantages of this method relative to the first method are that it halves the storage required to encode the span of the lane characteristics and that the actual distance covered by a certain characteristic is explicit, instead of being derived as in the first method. One disadvantage is that the actual start and end points are derived here, while they are explicit in the first method. Another is that the records would have to be ordered to know which portion of the link each characteristic covered. The second disadvantage is more significant, but if careful data management procedures are followed, it will not outweigh the storage benefit that results.

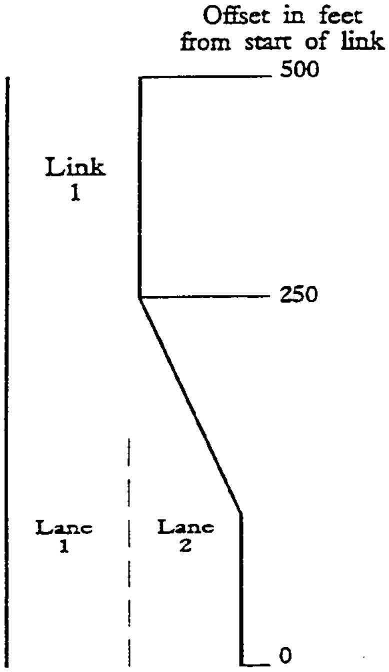
Another option deals with how the characteristic of the number of lanes is actually stored. This can be accomplished by storing the span of specific lanes or by storing the number of lanes as the phenomenon that occurs along a route, i.e. storing the offsets of the individual lanes rather than the offsets 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 lane number characteristic. 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 links the beginning and ending of the link. The second method forces discontinuity in the lanes.

The purpose of including lanes in the data model is to associate information with them for traffic modeling or route guidance. For the traffic modeling applications, the data associated with the lanes will be flow information. This may be from a monitoring system, or it may be derived based on the lack or presence of obstructions to flow. For route guidance, the important data will be simply the presence of the lane, what directions of movement are possible from it, and again whether the lane is obstructed. Therefore the table that contains the lanes will also require places to maintain flow information, turn possibilities, and the location of an obstruction. These attributes can be stored in the same way regardless of what method is used to indicate the changes in the number of lanes.

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 that is assumed and the 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. While there is a gradient of the complexity of this structure, this report will describe only three discrete 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.

Figure 3
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

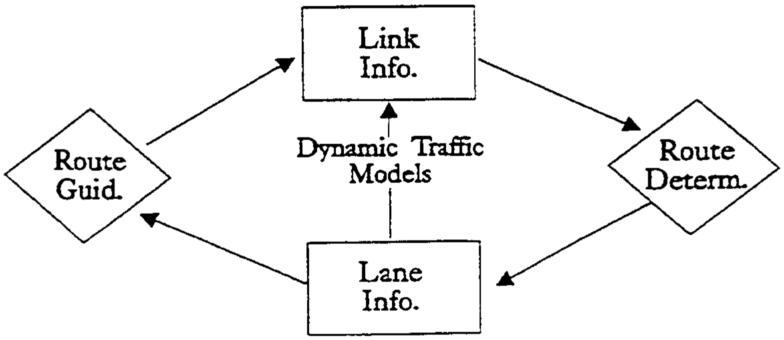
Option 2

Link Table

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

This figure shows the two options for storing the length of a given number of lanes along a link. Option 1 stores the start and end offsets from the beginning of the link of each continuous number of lanes. For example, the two lane portion of link 1 begins at an offset of 0 and ends at an offset of 250. Option 2 stores the extent of the link covered by each continuous number of lanes. For example, the two lane portion of link 1 covers 250 feet of the link. In option 2, the order of the records is critical for the system to know where on the link each span of lanes occurs.

Figure 4
Flow of Lane and Link Information for Route Determination and Route Guidance Tasks



The first option for representing lane connectivity at turns is actually to ignore it. One could argue that for typical automobile drivers, the need for explicit lane connectivity in the database is not obvious. Route determination algorithms typically use impedances in the turn as a whole rather than impedances for each lane. Route guidance can then give directions based on the lanes along the links of the chosen route rather than on how the lanes relate to each other in a turn. This is possible in two ways. First, the system may operate with the assumption that a turn from a lane on a link will proceed onto an analogous lane on another link. For example, if turning from lane one of three (i.e. the far left lane of three), one will turn onto the far left lane. If only one lane exists, the turn will proceed onto that one lane. Specialized lanes such as middle lanes in a link that serve as left turn lanes in both directions from any point in the link pose a problem for this approach. The second method would give constant directions regardless of the lane position of the driver. For example, the route guidance system could give directions to get into the right lane which may be followed by the driver if not in the right lane already.

How would this approach function if a lane becomes blocked in an intersection or close enough to the intersection that it changes the lane connectivity in the turn? For safety reasons it is highly unlikely that a lane within the turn would become deliberately blocked without the appropriate lanes on the links leading into the turn being blocked. If the lane is obstructed due to an event such as an accident, this can be handled by instructing the driver to get in the appropriate lane before the turn. The same method can be used if a lane is blocked in a link close to the intersection, so that one lane is not available in the turn.

The advantage of this approach would be the clear storage space benefits. The entire lane connectivity table which would invariably be extremely large could be eliminated. One disadvantage is that turn restrictions for each lane must be encoded in the database to provide proper guidance. Another disadvantage is that cases of connectivity of special lanes such as bus lanes or carpool lanes would not be maintained explicitly and could lead to problems. in addition, it may be problematic in cases where several lanes merge or where they converge on a highway, but they do not merge, i.e. the far left lane from one road may no longer be the far left lane on the next road.

Even if lane connectivity at turns is not encoded in the database, directional restrictions on lanes, e.g. left turn only, must still be included in the database. This is the second possible method for storing lane connections. 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.

Navigation/Routing with Lanes

The previous discussion implies that the routing application will be performed on the link structure and not on the lanes. This assumption is true to the extent that the routing applications use traditional algorithms that traverse each link to a node and use each node as a decision point for the next step in the route. It means that the route determination will be based on the impedance information for the links and turns as a whole rather than for individual lanes. Once the optimal route is selected, the system can refer to the lane information to guide the driver through the route. If this is the case, the most efficient approach in the planar data model is probably the second approach. It is a balance between storage requirements and data necessary for route guidance.

In this case, there would be a circular movement of information. The impedance values for the links are often based on values for individual lanes and generated with dynamic models of congestion along a link. Thus the lane information would be used to generate link information. The link information would then be used to determine the optimal route on a link-by-link basis. Once the optimal route is found, the links in the route would be passed to the route guidance system that would refer to the individual lane information in each link, providing detailed directions through the route (see Figure 4).

While the data structure for this type of model would be extremely large, a larger table can be broken into parts through table normalization. This would preserve the hierarchical relationship between the link as a whole and individual lanes on the link. It would be possible to assign attributes to an entire link in one table, attributes for lanes in another table, and have the lane table referenced by the link table.

It may be desirable to perform routing using individual lane information. This may even be advised when a special use lane exists, and the routing task is for the client of the lane. For this detailed level of routing, land topology is most appropriate. This routing is conceptually similar to the traditional routing techniques in that it would look at impedance on the lane in the direction of flow and impedance in the turn, but it has the added question of possible lane changes.

Turns from a link

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. In other words, if a destination of interest is on the left side of a link relative to the direction of travel, is it possible or legal to cross over the oncoming traffic to get to the destination? 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 modeling 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 merely 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. In this case the links represent a behavioral separation. 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 simply 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 for modeling 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 modeled explicitly 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 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, modeling them 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 modeling 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.

E. Development of Prototype

The following, section describes the prototype of the data model and its function. It reiterates the objectives of the prototype and describes the work completed so far. It also includes a brief statement of the specific steps that are necessary to complete the prototype.

Prototype Objectives

The purpose of the prototype is to demonstrate the effectiveness of the planar and nonplanar models to support IVHS functions. It will illustrate the feasibility of implementing the solutions to the special issues discussed in this report related to building a database for IVHS. This phase of the prototype will concentrate on the planar model and its efficacy. It will construct a planar database for a portion of the City of Santa Barbara using data provided by ETAK, Inc. and Navigation Technologies. Thus the prototype will also allow an opportunity to evaluate the quality of these third party data sets.

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.

Progress To Date

So far, we have imported both the ETAK and NavTech data sets. We have identified the prototype area and have created data sets of this prototype area from the NavTech and ETAK data.

We have read in the data from both companies and isolated the prototype area for both data sets. We have performed more processing on the NavTech data set than on the ETAK data set. This is has several causes: first, the NavTech data arrived considerably earlier than the ETAK data, and second, the NavTech data quality appears to be better in some respects which has made querying a subset of the data easier. In both cases, we have spent considerable time becoming knowledgeable about the data to understand the nature of the specific data contained and how they have been coded. Examples include the data model used to represent certain instances in the road network, whether turn restrictions are stored and the method for storing them, and the exact scheme for storing addresses along links.

We have performed the following processes on the NavTech data:

- creating the structure to support address matching
- geocoding the points of interest included in the NavTech data,
- creating a pure planar structure from NavTech's data model which is not entirely planar (NavTech does not store intersections/nodes where physical intersections do not occur, i.e. any overpass along 101),
- creating the turntable structure,
- entering NavTech-provided turn restrictions (the new NavTech database does include posted as well as logical turn restrictions),
- entering physical turn restrictions for nodes that were added to model a planar structure.

In addition to processing the digital data, we have spent considerable time evaluating the possible methods for representing lane information within the planar model and the techniques for implementing these methods using Arc/Info and the data that we have. The lanes will be incorporated using the offset method described in this report.

Future Steps

Among the steps to be performed are:

- entering all of the turn restrictions including any remaining physical restrictions as well as one-way related restrictions (i.e. not being able to turn from one road onto another because the second road is one way in the other direction),
- entering impedances for turns based on presence of signals or stop signs in the intersection[1],
- writing AMLs to perform simple routing and guidance functions,
- report on the effectiveness of the planar model to support the information needs specific to IVHS tasks.

The results from the prototype will also enumerate instances of road and turn configurations that require special attention for implementing the data model. These will include both common and pathological cases. Among the instances that will be considered are:

- subcases of intersections:
    - one or two left turn lanes
    - one or two right turn lanes
    - multi-directional turn lanes: e.g. left and/or forward progress allowed, right and/or forward progress allowed,
- entrance and exit ramps and lanes on and off highways,
- turn lanes that do not intersect with another link (e.g. into shopping center parking along a link),
- bus, bike, or carpool lanes,
- lateral obstructions along part of a link (e.g. concrete barriers restricting left turns).

---

[1] An alternate approach would be to classify the intersection as to type and condition. A table associated with each intersection type and condition can be used to store delays and impedances. This would obviate the need to maintain data for each intersection.

IV. DISTRIBUTED DATABASE ISSUES

A. Introduction

This section of the report continues the research into issues related to distributing the Caltrans navigable database. It begins with a discussion of current geographic information system (GIS) architectures, progresses to an evaluation of these architectures in supporting a distributed navigable database, and finishes with a proposed prototype system that implements a subset of presented requirements.

In the distributed database issues section of the preliminary progress report on the Caltrans navigable database (NCGIA, 1993), a broad set of requirements was presented that drive the general necessity for a distributed database. Also included in this report was an examination of a selection of database management systems (DBMS) and their potential to support a transparently distributed database. These standard DBMSs do not inherently provide the ability to store basic geographic entities that arise in the development of a map database (i.e. roads, intersections, census tracts), nor do they offer spatial operators to query these entities (i.e. distance, shortest path, adjacency). This implies that these standard DBMSs are unable to handle navigable queries like: "find the shortest path to the nearest on-ramp."

The Caltrans navigable database is primarily map-based, and the aim of this next stage is to move toward evaluating systems that inherently support the unique requirements for managing, manipulating, and analyzing a map database: geographic information systems. Most commercial GISs are based on a DBMS similar to the ones discussed in the first report, and this shift in focus merely serves to elevate the discussion of distributing a navigable database into a more relevant light. This relevance arises primarily from the notion that distributing a map database involves issues above and beyond those of distributing a non-spatial database. This section of the report presents a framework for organizing and evaluating contemporary GIS architectures, compares their relative merits and drawbacks in supporting a distributed navigable database, and presents a prototype system that serves as a spring board to discussion regarding Caltrans' requirements.

B. GIS Architectures

This section describes three dominant architectures for GIS that have emerged, in addition to reviewing some of their relative merits and drawbacks. It is intended to lay the groundwork for a subsequent section that reviews some specific examples of these approaches and their relevance to the Caltrans project.

A GIS can be thought of a collection of procedures or operators that allow one to select, process, and update elements from a spatial data structure or spatial database, where the ultimate task is to model some aspect of a spatial reality (Geuvara, 1992). The elements of a GIS are generally embedded in a two dimensional map-based space, and three primary architectures (see Figure 5) have evolved to handle the issues associated with managing a map-based or spatial database (Vijibrief & van Oosterom, 1991). They appear in the GIS literature under a variety of names, but for the purposes of this report the three approaches will be referred to as the dual architecture, shell architecture, and integrated architecture. The following three sections define these three approaches and compare their respective positives and negatives. It is important to note that not all GISs fall cleanly into one of the three categories, but the classification scheme holds for many GISs and does provide a valuable basis for discussion. In addition, these initial positives and negatives are broad-based, and a more specific discussion of the relevance of these respective architectures to distributing the Caltrans navigable database is left to a subsequent section on distributed geographic database alternatives.

Dual Architecture (Geo-Relational)

The most common approach to designing a GIS involves separating each geographic entity into its respective spatial and thematic components and storing these two components in separate "dual" databases. Common examples of geographic entities in the context of a navigable database might be road segments, intersections, or points of interest. The spatial component of an entity, in this context, refers to its geometric representation (i.e. point, line, polygon). The thematic component can be viewed as a description of the object and its attributes. This is to say that a line entity in a map database can represent any number of linear real-world entities from roads to streams. The thematic component serves to give the geometric component an identity (i.e. a highway link with impedance, capacity, etc.). These two components are linked via a common identifier across the two DBMSs, and retrieving an object involves querying both subsystems through an integrated environment to compose the final answer. In most instances, the thematic component is stored entirely within a commercial relational DBMS (RDBMS) as tabular information. The geographic component, on the other hand, is stored in a proprietary database with its own unique internal access and storage methods. This approach is also commonly referred to as the geo-relational approach, with ARC/Info being the most popular example. In this case, ARC is the spatial database, and Info is the relational DBMS used to store the thematic component.

Primary GIS Architectures

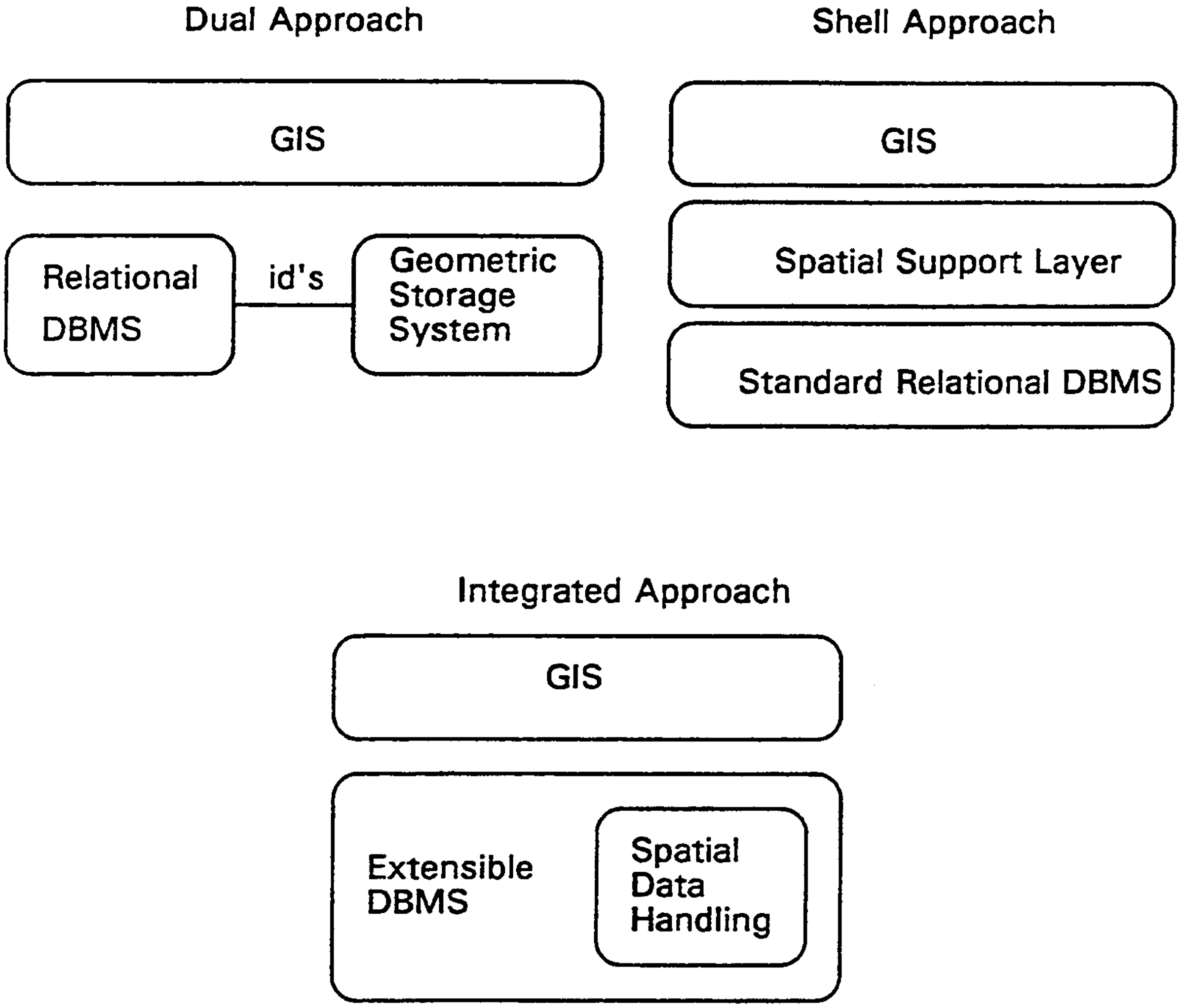


Figure 5. This figure shows three conceptual representations (Viljbrief & van Oosterom, 1992) of the three most common architectural approaches to building a GIS. In the dual architecture, geographic entities are separated into two components (spatial and thematic) and these two components are stored separately in two separate DBMSs (geometric and relational, respectively) and linked via a common identfier. In the shell approach, a spatial support layer is introduced on top of a standard relational DBMS to manage the spatial component of the data. This intermediary layer translates spatial queries into standard SQL to retreive the geographic information from a standard relational DBMS. In the integrated approach, an extended or object-oriented DBMS is required. In this case the management of spatial data is added directly at the DBMS layer.

This architectural approach usually allows integrating the spatial database with a selection from any number of commercial off-the-shelf (COTS) relational DBMSs (i.e. Oracle, Informix, etc.) through some type of database integrator or gateway. Consequently, the primary differences between individual GISs at the DBMS level exist within each vendor's unique implementation of the spatial database management system. This market, like many, is highly competitive, and although the concepts behind the design of these spatial database management systems are well documented, the actual systems are ultimately black boxes. System benchmarking of the capabilities, as well as the efficiency, is the accepted means of evaluating and comparing these systems.

Advantages

The main advantage of the dual architecture is that by storing the spatial component of the data in a proprietary fashion, the retrieval and management of this data can be made extremely efficient. In this manner, the individual vendors can fine tune their spatial database engines unconstrained by an underlying "outside" DBMS. Hence, the design of these spatial subsystems is the major arena for competition between GIS vendors. Another advantage is the utilization of a standard RDBMS to store the thematic component of the data. This allows users the freedom to design a portion of the database within the relational paradigm, and to utilize the benefits provided by the RDBMS vendors regarding querying, report generation, data management, and myriad other tools and facilities provided with these systems. Overall, the dual approach has been the driving force behind the growth in the GIS industry to date.

Disadvantages

Ironically, one of the primary disadvantages of a dual architecture arises from the same area as some of its advantages: the separation of the spatial data component from the thematic component across two DBMSs- This places the database in an awkward position regarding integrity, as entity components are linked only by a crosssystem id. RDBMS are designed to maintain data integrity in a concurrent transaction management arena. The DBMSs do not expect a critical portion of the database to be stored separately, and hence the overall integrity of the database is tenuously maintained though a unifying umbrella environment. In a single-user environment, the dominant model for GIS use to date, this problem is negligible. However, in a large-scale, concurrent, multi-user, transaction system similar to the one required to support a state-wide navigable database, this would pose limitations to the potential utilization of this architecture.

At this point, it is also not clear as to the limitations of the dual (,geo-relational) model regarding the handling of complex and composite geographic entities. This is to say that when one "buys in" to a dual architecture, one essentially agrees that the provided spatial data model is flexible enough to meet all future requirements. ARC/Info has recently been augmented with the facility to handle what Environmental Systems Research Institute (ESRI), the producers of ARC/Info, refer to as regions. Regions are modeled as composite collections of polygons. This exam le of a vendor addition to the spatial data model is the only way that the data model can be augmented. ESRI has managed to keep up with the spatial data modeling needs of the GIS community to date, but computer scientists are hard at work developing systems that put extensibility in the hands of the user. This will be covered more in the discussion on an integrated architecture. At this point, extensions to the geo-relational model continue to arise (Maguire et al., 1992) that intend to quell the criticism that the model is going to reach an end. As the georelational approach is a relatively mature architecture, it is receiving less attention in the research community at large notwithstanding the fact that it is still the dominant approach in the commercial sector.

Shell Architecture

The second approach to designing a GIS is the shell architecture. This approach is an attempt to overcome the drawbacks of storing the two components of geographic information in separate subsystems by storing all data completely within an RDBMS. In this architecture, the spatial data are separated into their basic elements (i.e. points, lines, polygons) and stored in related tables. To retrieve the information, relational joins are performed to reconstruct the required geographic entities. This is accomplished, and made efficient, by placing a spatial support "shell" on top of a standard RDBMS to handle geographic queries. The success of this approach, therefore, hinges on the intelligence of the design of the spatial shell. Popular systems that rely on this approach are System 9 (Penderson & Spooner), GEOVIEW (Waugh & Healy, 1987), and SIRO-DBMS (Abel, 1989).

Advantages

In the shell architecture, complete support for transaction semantics and integrity is assured at the level supported by the underlying RDBMS. To release the user from the burden of performing difficult queries directly in SQL, intelligent spatial query systems like GeoSQL are incorporated within the spatial shell level that have the benefit of being affiliated with the strength of the SQL standard. Also, all the benefits provided by the RDBMS vendor regarding tools and facilities are available at the supporting level. This means that a GIS built on a RDBMS that supports the transparent distribution of data would itself have a better chance of tapping the necessary resources over a GIS that adopted a dual architecture - that is, if the GIS vendor that provides the shell is interested in transferring these underlying RDBMS features to the user operating at the GIS level.

Disadvantages

The primary drawback of the shell approach arises from the shell that is placed between the GIS layer and the RDBMS. Coherent geographic entities must be decomposed into multiple parts and collapsed into tables, and this decomposition and recomposition of spatial entities can place efficiency limitations on the system as a whole. In essence, the indirect implementation of spatial index methods on top of a RDBMS is less efficient than if they can be implemented directly within the DBMS itself. In many ways the proprietary shell between the user and the underlying RDBMS has the same drawbacks as the dual architecture. The data model is established by the GIS vendor, and the system is essentially non-extensible. This may not be as great a disadvantage as it appears, as most GISs rely on a spatial data model that is robust and flexible enough to handle the demands regarding a navigable database. This is due to the fact that the map scale of a navigable database is such that the geographic component of the database is a link-node structure and the primary geometric entities are simply points and lines. The sophistication and uniqueness of the model for a navigable database is in the design and management of the thematic component (see prior section on IVHS data models).

Integrated Architecture

The third primary approach to developing a GIS is the integrated approach. This approach is the broadest and least exploited within the commercial GIS community. It is, however, a research focus both in industry and in academia. In this approach, the spatial data management and storage facilities are built directly into the DBMS itself. This precludes the use of a standard RDBMS as the foundation for this architecture, as the facilities for this approach are unsupported by these commercial systems. Instead, an extended DBMS or object-oriented DBMS is required. Another driving force behind this approach is the notion that users and application developers can extend the system with their own abstract geographic data types and behaviors. However, this freedom requires a high degree of expertise, and it is not expected that the average user will want to derive new geographic data types. In general, this is not a widely available option in the commercial GIS market, but it shows the most promise within the academic and industry research community as the GIS architecture of the future. Examples of this architecture include TIGRIS (Herring, 1987) from Intergraph, Gdotropics (Bennis et al., 1990), and GEO++ (Vijlbrief & van Oosterom, 1991). SMALLWORLD (Newell, 1992) also possesses some of the characteristics of this approach, and SMALLWORLD is a good example of a system that doesn't fall directly into one of these three architectures.

Advantages

One primary advantage of this approach is the flexibility to implement an unlimited amount of real world geographic data models. This is due to the extensibility of the architecture, as any number of geographic data types can be developed and modified along, with their behavior and relationships directly at the DBMS level. Another key advantage is performance. Implementing the data types directly at the DBMS level is extremely efficient, and the benchmarks that are being posted by some of the object-oriented DBMS (OODBMS) systems in retrieving and updating spatial data are impressive (Milne et al., 1993). This will be covered more in a subsequent section on geographic object-oriented data modeling and management systems.

Disadvantages

The primary disadvantage of this approach is the time and difficulty involved in implementing a spatial database management system on top of an extended or object-oriented DBMS. These are sophisticated DBMSs from a database management perspective, but from the point of view of an application programmer, the DBMS is responsible primarily for simply adding persistence to objects that are created in a standard programming environment (i.e. C or C++). This means that application development begins with code line number 1, and a quick skim though the functional reference manual of any standard GIS will testify that a large number of functions are required for spatial data input, management, display, manipulation, and presentation.

There is wide research on the part of academics and industry groups alike to move this approach forward, but at this point there is not a clear "winner" that utilizes the integrated GIS approach. For this reason, much of the development that occurs is application specific and done by programmers and DBMS experts. JHK's transportation operation center is a good example of a application specific system that is developed directly on top of an objectoriented database (JHK, 1993).

Object-Oriented and Extended Relational GIS

Due to the stated necessity for an object-oriented or extended relational DBMS in the desig of an integrated GIS, it is worth discussing some important concepts and developments within the spatial database research arena regarding this approach. This focus represents an entry into a large literature, and this section is only intended to begin the discussion of the relevance of object-oriented and extended relational DBMS regarding the Caltrans navigable database project.

As mentioned in the prior section, one approach to implementing an integrated GIS architecture utilizes an object-oriented DBMS. The term object-oriented appears most commonly in reference to a software design methodology (Booch, 1991), but, although it is used often in data modeling, it is still ambiguous when the term appears in this regard. As noted in the first progress report to Caltrans, the object-oriented model is based on the concepts of encapsulation, hierarchical classes, methods, inheritance, and polymorphism (NCGIA, 1993). According to Newell, it may be more appropriate to think in terms of degrees of "object-orientedness", depending on how many of these concepts of object-orientation are included in a design. Despite this lack of precisely defined rules, there exists a growing literature on spatial database design that involves concepts drawn directly from the object-oriented paradigm. A number of authors have described various approaches using these principles to constructing a GIS (Newell, 1992; Berrill & Moon, 1992; Egenhofer & Frank, 1992; Guevara, 1992; Gahegan & Roberts, 1988; Scholl & Visard, 1992), while others have applied similar concepts directly to spatial data modeling (Worboys et al., 1990; Worboys, 1992; Zhan & Mark, 1992; Herring, 1992; Clementini, 1991). Milne (Milne et al., 1993) recently published an example case study on the general use of object-oriented databases in geographic modeling.

One object-oriented approach by Newell in the design of SMALLWORLD GIS is an interesting case in point. In this system, Newell points out that:

> "In our work, the fundamental persistent storage is tabular, but using the principle of encapsulation the table/record structure is made to look like an object data structure. At the lowest level, a table looks like an indexed collection, a record looks like a slotted object and a field is a slot. Tables understand messages about relational algebra."

This implies that one approach to object-oriented database design involves a higher-level conceptual level that essentially organizes fundamental objects that exist in a logical data model very similar to the table structure of the relational model. In short, it is still very efficient to store geographic information in a tabular form, but significant design variations exist in the overlying conceptual model used to organize the tabular information. This idea is reiterated by Guevara (Geuvara, 1992) in the following form:

> "The relational approach to spatial data handling falls under the general functional model (functionally driven inferences on spatial data primitives), while the object-oriented approach falls under the specific derived model (data and their operations are encapsulated to establish relationships between spatial primitives). It is important to understand that these two models are not mutually exclusive (i.e. a general functional model can be used to support a specific derived model). "

The point that the object-oriented model may employ underlying data structures similar to relational tables is significant. It reiterates the point made in the introduction of this section that the prior classification of GIS architectures is by no means absolute, and architectures exist that are hybrids of these classes.

C. Distributed Navigable Database Alternatives

This section shifts the attention to the potential for the previously delineated three GIS architectures to support a navigable database. Each section covers an example implementation utilizing one of the architectural approaches and examines its positives, negatives, and potential to meet Caltrans' requirements as presented in Caltrans Progress Report I (NCGIA, 1993). Although specific commercial off-the-shelf software products are reviewed in this section, they are intended primarily as examples. Thus the comments made in this section should not be construed as specific recommendations to Caltrans.

The recent increase in IVHS activity is placing a formidable challenge in front of the commercial GIS world, and vendors are attempting to move toward a central role in serving the spatial database needs of this emerging market. However, the requirements of IVHS place a family of demands on the commercial GIS field that it was not historically designed to meet. Three of these are real-time data management, concurrent access, and data distribution. Although the historic view of a GIS involves a single user and a single database, this view is gradually eroding as vendors move toward meeting these new demands presented by markets like IVHS. Unfortunately, at this point, it is not clear whether commercial GISs will rise to meet these needs sufficiently, or whether special purpose systems designed directly on top of a general DBMS will fill the IVHS niche. As mentioned in the section on dual GIS architectures, many vendors are tied to a proprietary spatial database and a standard RDBMS. This may impose limitations on their ability to meet the dynamic needs of IVHS. On the other hand, it will take time before vertical market IVHS application environments emerge that rely on a leading-edge object-oriented or extended DBMS. This trade-off between utilizing commercially available technology over developing an IVHS system from the ground up represents one of the single largest questions regarding the development of a state-wide navigable database. It is the NCGIA's recommendation that a project of this nature should attempt to maximize the use of commercially off-the-shelf technology over reinventing the wheel whenever possible.

This section raises some of the currently available avenues for implementing a distributed navigable database within the context of the three architectures presented in the first section: dual architecture, shell architecture, and integrated architecture.

Distributed Navigable Database Requirements

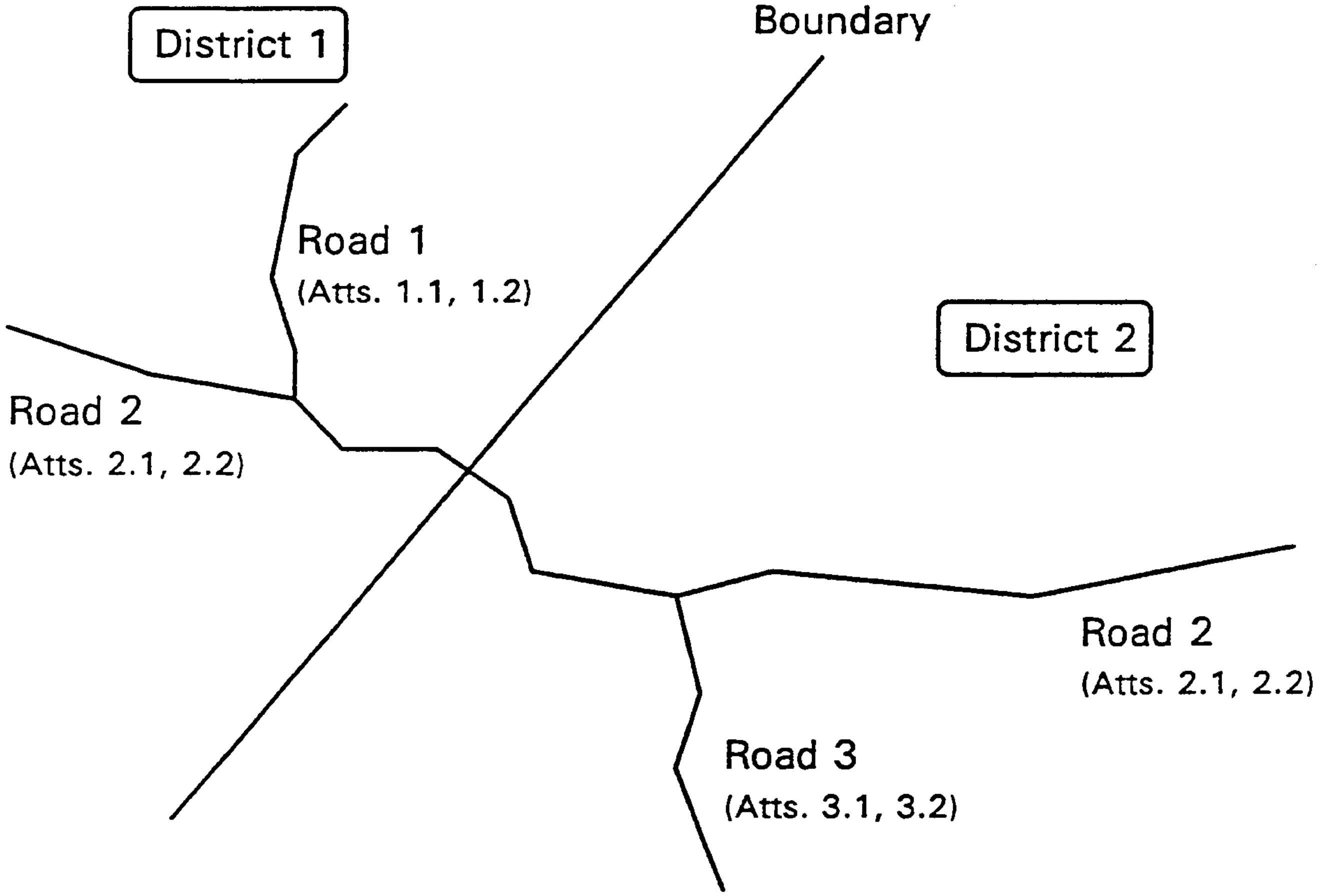
Before discussing a subset of specific directions to implementing a system capable of supporting a navigable database, it is worth reviewing the general requirements that are inherent to the Caltrans project in the light of an additional six months of research.

The primary purpose of the Caltrans navigable database is to support the development of IVHS applications, and the development environment should possess the capabilities to manage the complexity associated with these applications. One example of complexity is the requirement to support real-time applications. This means that the system should have some facility for acquiring data in real-time, updating the display in a reasonable time, and defining triggers that invoke actions as conditions within the database are met.

Secondly, local autonomy is a primary driving motivation for the development of a distributed database. The Caltrans navigable database is to be partitioned both spatially (see Figure 6), where each district controls the data that pertains to its local road network, as well as by individual road attributes (see Figure 7), where complementary departments responsible for different services regarding a district's road network control the attributes relevant to their department. In this context, control is a synonym for autonomy and refers primarily to a site's ability to set its level of data sharing as well as maintain the responsibilities of updating data and overall data integrity. It should be noted that the term control does not necessarily imply that the data is physically located locally. This point is raised to clarify the fact that important subsets of local autonomy can be satisfied in a system that is entirely centralized, as autonomy is granted chiefly though update and maintenance rights to portions of a database. However, there are other requirements within the definition of local autonomy that preclude the centralization of a database. These are the concepts of site equality and continuous operation. Site equality assures that no site has dominion over another site, and continuous operation enables a site to operate independently of other sites should any one of the other sites experience a crash or any form of downtime. In short, important subsets of local autonomy can be met satisfactorily without a transparently, physically distributed database There are in many cases efficiency and integrity gains to be garnered by centralizing a complex spatial database, and it is a management decision as to how strictly the local autonomy requirement is to be taken. In fact, Stonebraker's (Guptill & Stonebraker, 1992) BIGFOOT I prototype in the Sequoia 2000 project is testimony that some of the forefront research in managing large object databases adopts centralization as a solution.

Juxtaposed to the requirement of local autonomy is the goal of data sharing. As part of its intermodal transportation plan, Caltrans realized the need for cooperation and coordination between Caltrans districts, and other cities and local governments, to achieve a statewide navigable database. The Caltrans system should allow sites transparent access to data controlled by other sites through distributed query processing. This transparency should insulate the application programmer at a site from the details of the data fragmentation (fragmentation transparency) as well as the location (local transparency). In short, if the data is physically distributed, the database should appear available in its entirety to any given site as though it were stored locally.

Horizontal Partitioning (geographic or spatial partitioning)



Original Complete Table

Road	Att. 1	Att. 2
1	a	d
2	b	e
3	c	f

so District 1 controls ...

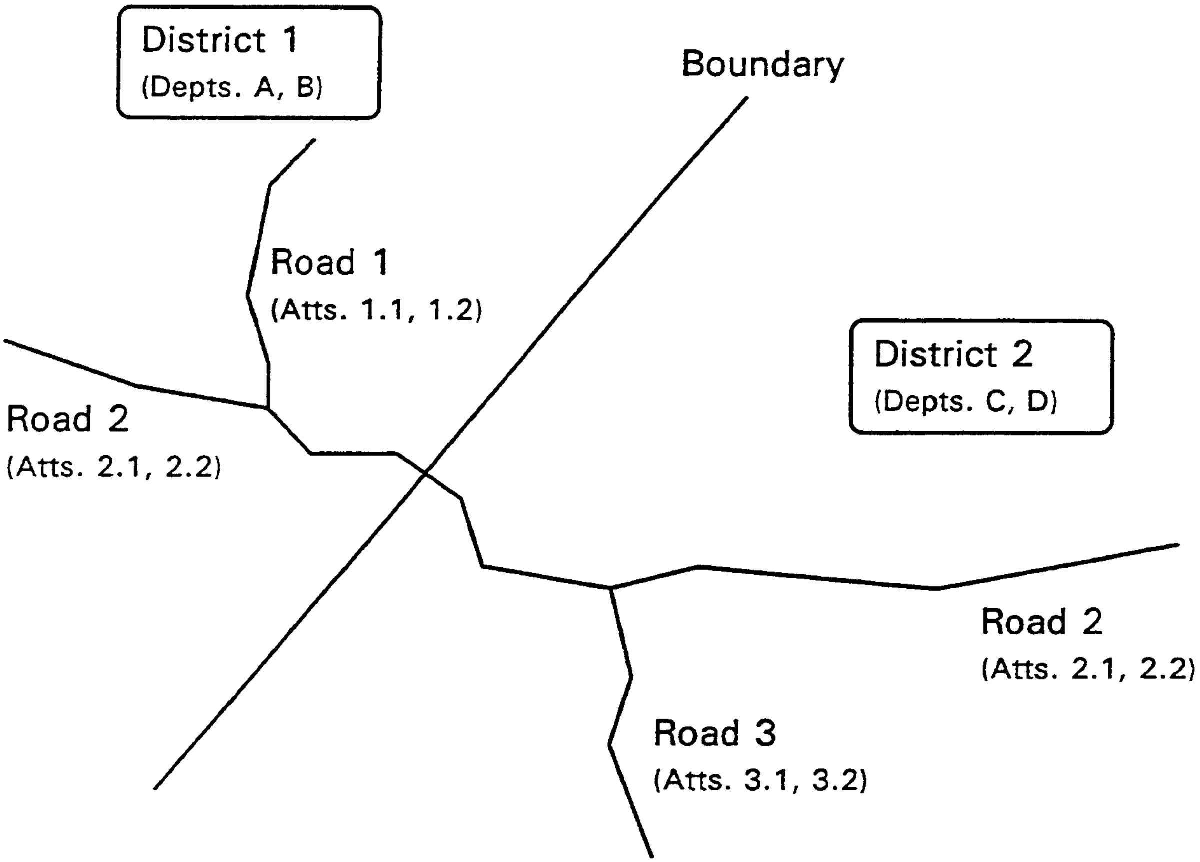
Road	Att. 1	Att. 2
1	a	d
2 (part)	b	e

and District 2 controls ...

Road	Att. 1	Att. 2
2 (part)	b	e
3	c	f

Figure 6. This figure shows the relationship between horizontal partitioning and its equivalent map-based representation. This does not necessarily imply that the table is physically separated into two tables, but only that each district "controls" its portion of the navigable database.

Vertical Partitioning (attribute or columnar partitioning)



Original District 1 Table

Road	Att. 1	Att. 2
1	a	d
2 (part)	b	e

so Department A in District 1 controls ...

Road	Att. 1
1	a
2 (part)	b

and Department B in District 1 controls ...

Road	Att. 2
1	d
2 (part)	e

Figure 7. This figure shows the relationship between vertical partitioning and its equivalent map-based representation. This does not necessarily imply that the table is physically separated into two tables, but only that each district "controls" its portion of the navigable database.

Envisioning a database partitioned across an unknown number of geographically separate sites immediately raises the question of efficiency. Many of the Caltrans navigable database application requirements involve GIS functionality, and a map interface is essential. The primary method for achieving the efficiency necessary for a map interface in a distributed database system is replication transparency. This means that readable copies of portions of the database exist at sites other than the one that stores the original copy. The solution should allow the time scale of update of the replicated copies from the original to be specified by a system administrator at the site containing, a copy. Though this type of transparency is preferred, it need not be mandatory in the distributed solution, as alternative means exist for updating replicated copies through application-driven data-subscription.

Implicit in any database solution is the requirement of data integrity. In a navigable database, this requirement becomes significant, as applications dependent on accurate routing are often miss ion- critical. The Caltrans distributed database system should allow the specification of distributed integrity constraints. This means the system should allow some mode, other than at the application level, for enforcing relationships between data stored at different sites. This would relieve the system administrator from having to continually verify the integrity of data at the application level.

Commercial Off-the-Shelf GIS - Dual Approach

Most of the leading commercially available GISs rely on the dual architecture. This section reviews one that the NCGIA is familiar with: ARC/Info.

ARC/Info from ESRI Inc. is a leading GIS package that relies specifically on a dual architecture, It is a mature product with a large user base and a history of meeting the needs of the GIS community. Currently, there is ongoing research at ESRI directed at meeting the unique demands of IVHS applications. One recent addition involves the implementation of a real-time (RPC) port that allows ARC/Info to receive attribute data in real-time and refresh the map display at a fixed interval. Another project, ARC/Storm, is intended to increase the data sharing capabilities of the system by moving the environment into a client/server arena. There is also a host of other projects related to transportation that involve interfaces between ARC/Info and popular transportation modeling packages (Le. TRANPLAN). In short, ESRI's direct interests are in meeting the needs of the IVHS community, and it is worth examinine the relative positives and negatives of this approach to implementing a distributed navigable database.

Positives

Clearly the most significant benefit to the ARC/Info approach is its richness in GIS functionality. In this sense it is a complete system that allows for the input, management, manipulation, analysis, and output of spatial, thematic, and temporal information. ESRI's experience in meeting the needs of the spatial data community for over 10 years has resulted in a wealth of experience in areas that traditional DBMS developers entering the IVHS market may not have encountered. Secondly, its fundamental spatial data model is capable of supporting the proposed "Ontario" model (NCGIA, 1993) chiefly via its dynamic segmentation and turntable facilities. Also, its network module provides functionality that would prove useful in building the fundamental building blocks for primitive IVHS tasks (i.e. shortest path). Lastly, there is high degree of expertise available in the labor market for building the necessary infrastructure to support the use of ARC/Info.

Negatives

One of the significant drawbacks to moving ARC/Info into supporting a large-scale distributed database is its underlying proprietary database and subsequent relationship with the relational database Info. These DBMSs are not highly regarded in the database community at large due to their limited ability to match the capabilities that are emerging from the next-generation object-oriented and extended-relational DBMSs. The ARC/Info environment has a smooth way of shielding the user from the limitations of the underlying cooperating DBMSs, but, in short they are awkward and inefficient systems by 1994 DBMS standards. It is difficult to envision ARC/Info supporting a state-wide, cooperative, distributed navigable database with upwards of thousands of users. There are however essential roles that ARC/Info might play in the system as a whole. Data input, editing, analysis, and presentation are four areas where ARC/Info has a lot of experience, and it would not be beneficial for a special-purpose IVHS system to embark on adding these management and display functions to the navigable database. This is the approach of Navigation Technologies in the development of their road network database and transportation application.

Secondly, application development is limited to the Arc Macro Language (AML), although there are available interfaces to outside programming, languages (i.e. C). AML is a proprietary, limited language that is generally not intended as a base for developing complex applications on the order of those required for IVHS (i.e. ATIS).

Conclusions

Overall, ARC/Info is ideal for developing and refining the definition of the Caltrans data model along with illustrating sample IVHS tasks for this project. It establishes a base from which to rapidly prototype a model system, along with fueling the ongoing debate between the trade-offs of established GISs and next-generation DBMSs. With the advent of ARC/Storm, ARC/Info also has the potential to play a valuable role in experimenting with distributed access to a navigable database. As the primary means for ARC/Info to communicate with other systems is via file transfers, it is not sufficient for experiments in transparently distributing a spatial database among a set of sites.

Secondly, it is also a powerful system for the input, editing, analysis, and presentation of geographic information. In this sense, it has the potential to play a valuable role in working together with another system that may handle the DBMS requirements of a state-wide, distributed, navigable database. The system is not, however, a stand-alone complete solution to meeting Caltrans requirements.

ARC/Info and Oracle

One significant addition to ARC/Info is its gateway to an outside RDBMS (i.e. Oracle) that allows ARC to access attribute data regarding geographic features that are stored in an external DBMS. This represents an alternative to distributing the navigable database that utilizes the capabilities of Oracle 7.0 for distributing a relational database. In this model, there would be an entire copy of the road network at each site, but the attribute data regarding the network could be shared among Oracle servers at each site (see Figure 8). In general this alternative has many of the positives and negatives of ARC/Info discussed in the prior section. However, this introduces the question as to the degree of importance involved in distributing the spatial portion of the database or the road network. In this arrangement a universal copy of the network would have to be assembled periodically from the changes made by the individual districts and redistributed to the districts in the form of a complete network. This is not as elegant as a system that allows for the transparent distribution of the road network, but it foregoes the integrity and efficiency problems that arise when a cohesive spatial database is fragmented across sites.

Oracle 7.0 - Shell Approach

As mentioned in the first section of this report, the shell approach involves developing a spatial support shell on top of a robust RDBMS that serves as an intermediary between a GIS and the underlying RDBMS. As noted, there are examples of commercial GISs that rely on this architecture (i.e. System 9), but this section focuses on developing, a special purpose environment directly on top of the RDBMS Oracle to specifically meet the requirements of a distributed, navigable database to support IVHS. It is intended that this alternative would facilitate tapping all Oracle’s resources for DBMS management directly rather than relying on a proprietary spatial shell between the RDBMS and an IVHS application. This, of course, represents a significant amount of programming effort, and this point as well as other issues are covered in the following sub-sections.

ARC/Info and Oracle Solution

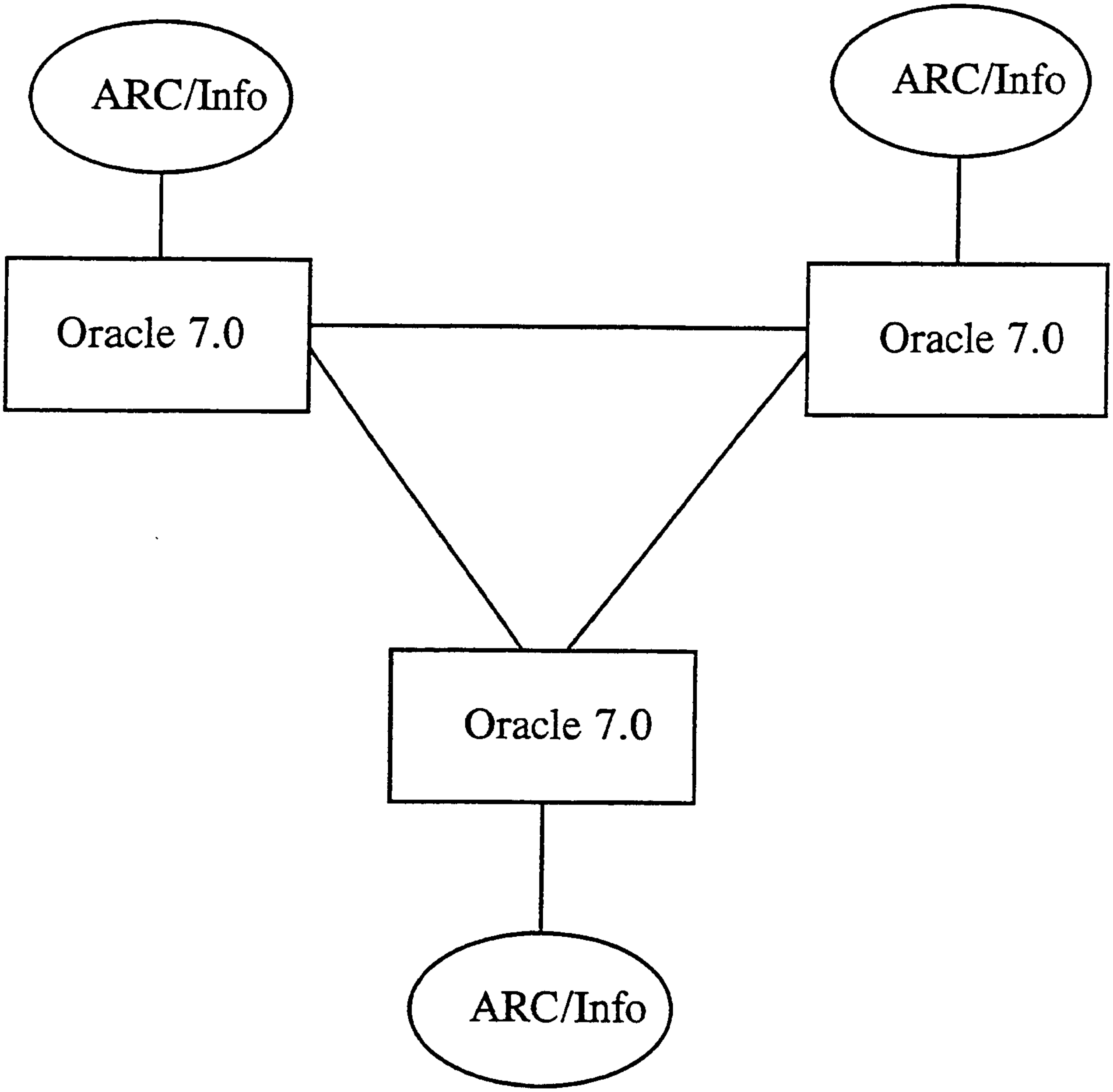


Figure 8. In this arrangement, a duplicate copy of the entire road network is stored at each of the three example sites. Attribute data is stored externally to ARC/Info by utilizing ESRI's database integrator and Oracle. Oracle 7.0 supports transparently distributed tables, and the connections between Oracle sites represent this facility. It should be noted, that this arrangement is theoretical and the difficulties involved in implementing it have not been explored in this contract.

Positives

With the release of Oracle 7.0, Oracle Corp. has made an initial launch at developing; and supporting the requirements of the distributed database market. Their goal of a transparently distributed relational database indicates that the commercial RDBMS market is beginning to adopt a body of concepts that has existed for some time as primarily academic DBMS research. This functionality has emerged too recently for any GIS vendors to have capitalized on it, but it introduces the potential to build a navigable database that relies on these features for data sharing; and update.

The relational model has many positives, most of which were covered in Caltrans Report I (NCGIA, 1993). In addition to these there is a high degree of experience in storing spatial databases in the relational model on the part of GIS vendors, academics, and the labor force. The standards of design and query are also established with formalized methods like relational calculus, algebra, and SQL. This approach also allows development of the system and its applications to be accomplished in a non-proprietary language (i.e. C, C++), as Oracle has a wide variety of development environment alternatives and tools.

Negatives

The difficulties of storing and retrieving complex spatial data structures in tables are well-known, as the relational model has existed for some time. This approach may lead IVHS developers down the same path that the shell GIS industry has traversed over the last 10 years. It is an approach that requires a large investment in the relational model and assumes that the dominant DBMS model of the past will remain the dominant model of the future. There is no doubt that tables are an extremely efficient storage means for non-spatial data, but with the rapid ascension of the extended relational and object-oriented DBMS market, it is in many ways a doubtful prediction that the purely relational approach will emerge in the end as the "best" way to store and retrieve spatial data. It is, however, possible (see prior section on object-oriented and extended GIS), that the relational model will emerge as an underlying support for a higher-level object-oriented framework.

Conclusions

This option allows the design of a special-purpose system that is tailored to handle the requirements of distributing a navigable database. However, IVHS applications are complex real-time applications that stress current database and display technologies, and current IVHS efforts appear to be moving toward object-oriented and extended relational databases to overcome this limitation (the Advance Project and JHK's TOC have both adopted Versant as the underlying DBMS to support their IVHS applications). This approach would involve a commitment toward developing a prototype to explore the possibilities for a relational navigable database.

Versant - Integrated Approach

As mentioned in the first section of this report, the integrated approach is the approach receiving the most attention in the GIS research community, chiefly because it shows the most promise as an elegant means of creating an extensible, efficient, spatial data management system capable of supporting concepts expected from a DBMS in general in 1994 (i.e. database distribution).

Versant was noted to be a leading object-oriented database management system that is currently available in Caltrans Progress Report I (NCGIA, 1993). It is not a spatial DBMS nor a GIS, but it is designed to support complex application development in the CAD, GIS, CASE, and IVHS arenas. There are GIS vendors that are investigating Versant as an underlying platform, but for the most part this option represents developing a system capable of supporting IVHS application from the ground up. This is the approach of JHK and associates in their implementation of the Transportation Operations Center (JHK, 1993).

Positives

Versant represents a sophisticated foundation for the development of a transparently distributed, navigable database, as it supports a rich degree of alternatives to implementing a complex geographic database where the integrity of individual geographic entities is preserved. This preservation of integrity is due to the fact that entity components are not divided across two databases, but are instead encapsulated together along with their "behavior". It is possible to develop the base spatial support module, as well as subsequent IVHS applications in C++. This is important, as it overcomes the impedance mismatch in systems where programmers must switch back and forth between an application language and a database language in their code. Therefore, the end-result code has a longer lifespan and higher value than if development was to be done in a proprietary GIS macro language or a DBMS-specific language. Versant is benchmarked much faster than relational databases at managing complex spatial objects (i.e. a road network), as the spatial data models can be stored in their native form over collapsing them into tables. The support for a scalable architecture that allows new users to seamlessly join the system with little to no effect on existing users is also a major plus in the development of a transparently distributed, navigable database.

Negatives

Perhaps the largest negative of this approach is that it requires a subset of basic GIS development from the ground up. Examining the functionality in a standard GIS for data input, display, management, manipulation, analysis, and output is an indicator of the endless amount of features that an IVHS system might require. A system of this type would effectively have to travel the same path as many GIS vendors have over the last 10 years, although not to same degree of generality as the application arena (IVHS) is more specific.

Conclusions

Versant is capable of handling the demands of Caltrans requirements for a very large distributed navigable database to support IVHS, although the development demands would be large. All development can be accomplished in C++ which has suitable long-term value. Benefits like Versant's scalable architecture indicate that database distribution is high on Versant's list ofimportant DBMS concepts.

It is possible that the best path for supporting the Caltrans navigable database project would involve developing under an OODBMS like Versant, but farming as many tasks as possible in the interim to a GIS like ARC/Info. This is to say that a file transfer between the Versant system and ARC/Info could be developed that would allow some of the database construction, manipulation, analysis, and presentation to be accomplished in ARC/Info, but with the idea that the final target platform is Versant. If possible, a reverse transfer without loss of too much information would also prove useful in updating and analyzing the evolving database. In short, a means of cooperation between a developing OODBMS platform for IVHS and an existing GIS would be a valuable union until the GIS industry itself evolves to directly handle the demands of IVHS. The goal is still to utilize commercial off-the-shelve technology to its fullest and essentially minimizing the amount of development using a system like Versant.

D. Prototype

This section of the report presents a prototype GIS with distributed access that meets a set of the requirements outlined in the preliminary Caltrans report, while illustrating an important distinction between a physically distributed database and a logically distributed database.

GIS as a Central Server

As mentioned in the preliminary Caltrans progress report, the primary goal in distributing a database is a satisfactory compromise between autonomy and sharing. This satisfactory compromise is somewhere between absolute autonomy where organization entities operate completely separately, and complete centralization where one organizational entity ultimately controls all aspects of a database. One approach to accomplishing the goal of distribed access to a navigable database involves moving a COTS GIS into a key position as a primary server. In this architecture, the GIS plays a central role in integrating data from a variety of disparate sources under one hood. These data may range from a basemap of the road network to the most frequently updated real-time attribute data regarding network conditions. This unified database can then be shared among application clients as a cohesive resource (see Figure 9). In this case, data that are distributed geographically can be brought together and shared among a collection of users or clients as a unified resource.

ARC/Info can be used to illustrate an example of how this can be accomplished. In this arrangement, ARC/Storm allows various clients to share a unified navigable database where attribute data regarding road conditions are refreshed from outside databases across a network. The real-time (RPC) port that ESRI has recently added would play the role of receiving data from the outside world. ARC/Info's database integrator allows these incoming attribute data to be stored in an external relational database like Oracle. They are then available for query as necessary.

Data Sharing (Client/Server)

ARC/Storm is ESRI's new client/server storage manger for use with ARC/Info. This product introduces a set of very appropriate features for the Caltrans project. Paramount among these is the ability for a collection of clients to share a single GIS database. This moves the distributed database solution closer to accomplishing the required data sharing goals. Clients essentially "log-in" to read the navigable database as though they had complete access locally. Although this satisfies some of the goals of sharing data, the question remains as to how to preserve the necessary autonomy of the separate clients that access the navigable database.

Autonomy (Distributed Rights)

In this proposed model for a distributed database, it is assumed that the physical distribution of the data is chiefly irrelevant. Instead, a form of logical data distribution is accomplished by distributing a commodity of extremely high value: update rights. It is the distribution of rights to the data that fulfills the goal of preserving autonomy. Each client is allocated rights and responsibilities for updating a portion of the GIS database. To guarantee the highest level of autonomy, this partitioning of rights can be non-overlapping, where clients are assigned update rights to features or attributes on a one-to-one basis. Two types of rights partitioning are supported: horizontal and vertical (see Figures 6 and 7). Horizontal, or spatial, partitioning involves assigning the update rights of a spatial region of the database to a given client. An example of this would be assigning a district the complete rights for updating all data regarding its roads. Vertical, or attribute, partitioning assigns clients update rights to individual attributes of the road network. One example of this would be assigning traffic volume rights to a given client. ARC/Storm is designed to support vertical partitioning of a database, as rights to each coverage can be assigned to a client. A road network and its associated attributes is an appropriate example of an ARC/Info coverage. A much higher level of granularity regarding rights is required for the Caltrans project. In the currently proposed solution, each map feature (Le. road segment, intersection) and each feature attribute (i.e. impedance, lane layout) will have its own update key. This key can be assigned to one or more clients depending on the degree of autonomy required. This provides a high degree of flexibility in distributing access to the database. With this degree of access granularity, it becomes possible to assign a client, for example, rights to update traffic impedance in a given spatial region. This is an example of combining horizontal and vertical partitioning, which may be mixed freely in a system that allows logical distribution of rights at a hi-h level of granularity.

The extension of "high granularity rights assignment" is accomplished by adding one column to the feature attribute table that is essentially an update key to the data. A screening of clients will be necessary to assure that only the clients with rights to an individual feature or attribute are allowed to modify it. One direction for this research prototype in the future might involve introducing a more peer-to-peer notion in the prototype solution where ARC/Info servers request information from each other as necessary via file transfers.

Prototype Solution

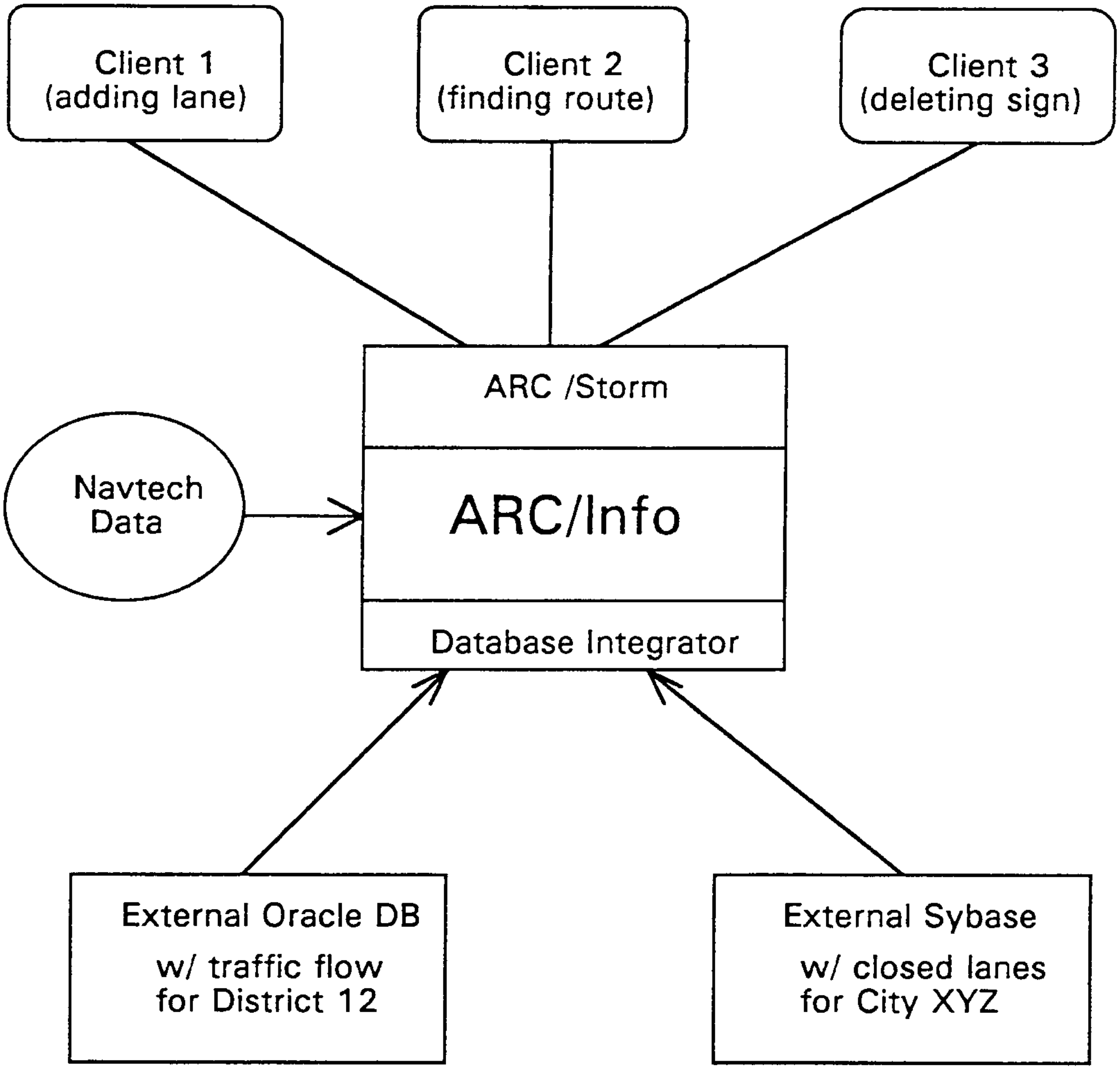


Figure 9. This schematic represents the prototype solution designed to enhance the discussion regarding distributing the database. The NAVTECH data is imported into ARC/Info and through much massaging is transformed into the final data model (see section of this report on IVHS data models). ARC/Storm provides a client/server flavor to the GIS that allows concurrent access to the database. Attribute data regarding the network is imported into ARC/Info at appropriate intervals to simulate real-time data acquisition.

E. Further Research

At this point it is evident that more research into object-oriented spatial data modeling and object-oriented design in geographic information systems would be a fruitful avenue given the attention that this approach is receiving in solving problems relevant to the development of a navigable distributed database in the research community at large. The initial report broached the emergence of OODBMS, and a shift toward complex spatial database modeling, and GIS architectures in this second report raises questions regarding the merging of object-oriented concepts and spatial database design. An initial review of this literature within this report has served to scratch the surface, and it is intended that a full review of this literature directly related to object-oriented spatial data modeling and geographic database design and its impact on the development of a distributed, navigable database is in the interest of this research group.

V. REFERENCES

Abel, D.J. (1989) SIRO-DBMS: A database tool kit for geographical information systems. International Journal of Geographical Information Systems 3 (2): 103 -116.

Berrill, A., G. Moon (1992) An object oriented approach to an integrated GIS system. GIS/LIS Proceedings, San Jose, CA.

Booch, G. (199 1) Object-Oriented Design with Applications. The Berjamin/Cummings Publishing Company, Inc.

Clementini, E., P. Di Felice, A. D'Atri (1991) An object-oriented conceptual model for the representation of geographic information. Symposium on Applied Computing, Kansas City, MO.

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

Davis, B, S. George, R. Marx (1992) TIGER/SDTS: standardizing an innovation. Cartography and GIS 19(5): 32 1 -327.

DOT (1993) IVHS User Services and Subservices. Department of Transportation, Washington, DC.

Egenhofer, M.J., A.U. Frank (1992) Object-oriented modeling for GIS. URISA Journal 4(2):3-19.

Federal Geodetic Control Subcommittee (1989) Multipurpose Land Information Systems: The Guidebook. Federal Geodetic Control Subcommittee,Washington, DC.

FHWA (1993) IVHS Architecture Mission Definition. FHWA, Washington, DC.

FHWA (1993) Technical Work Request: National Spatial Database and Link Identification for IVHS. FHWA, Washington, DC.

Frank, A.U., M.J. Egenhofer (1992) Computer cartography for GIS: an object-oriented view on the display transformation. Computers & Geosciences 18(8) : 975-87.

Gahegan, M.N., S.A. Roberts (1988) An intelligent, object-oriented geographical information system. International Journal of Geographical Information Systems 2(2): 10 1 -110.

Gordon, S.R., CW.H. Goodwin, D. Xiong (1994) Draft Final Report on Status of Spatial/Map Databases. Oak Ridge National Laboratory.

Guevara, J.A. (1992) On the design of geographic information systems procedures: the object oriented paradigm. Proceedings of the Twelfth Annual ESRI User Conference, PalmSprings, CA.

Guptill, S.C. (1992) The Sequoia 2000 approach to managing large spatial object databases. Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, SC.

Herring, JR. (1992) TIGRIS: a data model for an object-oriented geographic information system. Computers & Geosciences 18(4):443-52.

fHK (1993) Database Systems. TOC Component Evaluation, Document No. 30., Contract No. 04D922-PC. Emeryville, CA: JHK and Associates.

Laurini, R., D. Thompson (1992) Fundamentals of S patial Information Systems. Academic Press, San Diego, CA.

Magguire, D.M., G. Stickler, G. Browning (1992) Handling complex objects in geo-relational GIS. Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, SC.

Milne, P., S. Milton, J.L. Smith (1993) Geographical object-oriented databases - a case study. International Journal of Geographical Information Systems 7(l): 39-55.

NCGIA (1993) Progress Report - Caltrans Agreement 65T.115. National Center for Geographic Information and Analysis, Santa Barbara, CA.

NCHRP (1993) Implementation of geographic information systems (GIS) in state DOTs. Research Results Digest No. 191 (October), pp. 1-47. National Cooperative Highway Research Program, Transportation Research Board, National Research Council, Washington, DC.

Newell, R.G. (1992) Practical experiences of using object-orientation to implement a GIS. GISILIS Proceedings, San Jose, CA.

Penderson, L.O., R. Spooner (1988) Data Organization in System 9. WILD Heerbrug.

Peuquet, D.J. (1984) A conceptual framework and comparison of spatial data models. Cartographica 21:66-113.

Scholl, M., A. Voisard (1992) Object-oriented database systems for geographic applications: an experiment with O/sub 2/. Geographic Database Management Systems. Workshop Proceedings 103-37.

Siegel, D., P. Okunieff (1994) White Paper: Overview of Proposed Alternatives to Location Referencingfor IVHS User Services. Boston, MA: Viggen Corporation.

Vijlbrief, T., P. van Oosterom (1992) The GEO++ system: an extensible GIS. Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, SC.

Vonderohe, A.P., L. Travis, R.L. Smith, V. Tsai (1993) Adaptation of geographic information systems for transportation. Report 359, Transportation Research Board, National Academy Press, Washing on, DC.

Waugh, T.C., R.G. Healey (1987) The GEOVIEW design. A relational data base approach to geographical data handling. International Journal of Geographical Information Systems 1(2): 101-118.

Worboys, M.F. (1992) A model for spatio-temporal information. Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, SC.

Worboys, M.F., S.M Deen (1991) Semantic heterogeneity in distributed geographic databases. S[GM0D Record 20(4):30-4.

Worboys, M.F., H.M. Hearnshaw, D.M. Maguire (1990) Object-oriented data modelling for spatial databases. International Journal of Geographic Information Systems 4(4):369-383.

Zhan, F., D.M. Mark (1992) Object-oriented spatial knowledge representation and processing: formalization of core classes and their relationships. Proceedings of the 5th International Symposium on Spatial Data Handling, Charleston, SC.

VI. APPENDIX

This Appendix presents an experiment to investigate issues relating to integrating real time data (DGPS data) into the navigable database prototype. The objective of this effort is to establish a means of generating and disseminating real time data and to evaluate connectivity and system integration issues related to the navigable map database requirements.

The Testbed Center for Interoperability (TCFI) in cooperation with NCGIA/UCSB will evaluate the impact of real time data and related system integration issues on the navigable database design.

A Trimble differential GPS station located at NCGIA will be fitted with Spread Spectrum radio units. The Spread Spectrum stations will transport differential corrections to mobile and fixed stations. The differential station has a fixed known position, and the effect of multipath signals at the station will be evaluated and minimized.

A 386-based data bridge with different network and dial-up ports will be used to establish network connectivity to the Internet, where differential correction data may be retrieved through VIS-A-VIS or equivalent software. The field unit will be equipped with a Geolink mapping system, while the base station will be equipped with Penmetrics window-based software. Interface issues with the prototype database will be evaluated under the second phase of the project.