THE CONCEPTUAL UNIVERSE DATABASE SCHEMA: DESIGN ISSUES AND DECISIONS

A Technical Report

For The

North Carolina Department of Transportation

By

Dr. William Rasdorf

August 30, 1999

PROLOGUE

Significant progress has been made on the NCDOT Statewide Planning LRS and Database Project. Early in the project outstanding results were obtained with respect to the design of the Linear Referencing System and a report was written outlining the proposed base LRS.

It is important to note that with respect to the LRS, a solid technical solution was designed, one which fits NCDOT needs well. Technical problems of interest to the stakeholders were solved. But even more importantly, the LRS solution was obtained with consensus agreement. The process was one which also generated a new and deeper understanding of the broad and diverse information needs within NCDOT and the central role that spatial data plays in meeting those needs. The criticality of the LRS and its potential, even outside the stakeholder group, is clearly evident.

A conceptual database design for the LRS has been completed as well. A number of technical problems that were foreseen have been considered and solved. These results and solutions are outlined herein.

The ORACLE software is in place as is the ArcVIEW and ArcINFO GIS software. NCDOT is now poised to implement the solutions. The next steps are to:

1) Build the LRS as designed
2) Design the attribute database and interfaces
3) Build the database and interfaces

These tasks are nontrivial and will require significant resources to complete them. But the benefit to the NCDOT mission, in terms of increased automation, greater information sharing, improved reporting capabilities, and enhanced data analysis will be worth the investment.

1.0 INTRODUCTION

This section provides a general description of the overall objective of this report. Several terms are introduced, which will be used throughout the report. The section introduces the concept of a universe database and describes its implementation. The primary stakeholders involved in the development and use of the universe database are also introduced.

1.1 BACKGROUND

With the advent of advanced information technologies such as Geographical Information Systems (GIS), database management systems (DBMS), etc., the North Carolina Department of Transportation (NCDOT) has recognized the need to provide universal access to data. The NCDOT's main objective is to create an environment where data can be universally retrieved in a graphical GIS format or in a non-graphical database format. But before any work in the centralized database design could be accomplished a number of issues needed to be resolved.

The NCDOT began a three-part study that involved: (1) determining the most effective ***linear referencing system (LRS)*** for NCDOT's needs; (2) database design, processing, and development with attention given toward the types of data analysis functions performed by NCDOT, standardizing data terminology, and determining the shared data needs of the participating business and engineering units; and (3) providing recommendations for the development and standardization of attributes for the database. The first part of this study, which seeks to determine the most effective LRS scheme for NCDOT's needs, has been completed. This report is concerned primarily with the latter two tasks outlined above.

The ***Statewide Planning Branch*** desires to efficiently combine tabular and spatial data into an integrated system to support queries; applications; data entry and maintenance; and report generation, to the extent possible, from both the GIS and the DBMS. The design and development of a data and information management system that achieves these objectives is sought. This report looks at the spatial model and identifies and addresses key problems that could impede the design.

The NCDOT faces a significant challenge in its attempt to redesign the Planning Branch databases. The purpose of doing so is to make the databases more broadly available to other DOT Units and outside users as well. Additionally, a critical need is to make the databases more internally available, efficient, and useful. In short, NCDOT's main objective is to create an environment where databases can be universally retrieved in either a graphical GIS format or a non-graphical database format.

This report focuses on issues related to the design of the universe database schema. It presents a description of, and solution for, a number of the problems that emerged during the design. It also presents a number of the final database schema design decisions for the universe database.

1.2 PURPOSE

The purpose of this report is to present the solutions to technical problems associated with the development of a universe relational database model for NCDOT universe data. The universe database is designed to meet the overall needs of a specified set of individual DOT units including ***Statewide Planning (Traffic Surveys, Forecasting, and Planning), GIS, Traffic Engineering, and Pavement Management***, hereinafter referred to as the “stakeholders. It must be able to meet all of the data demands placed upon it from each of these units. A relational database design is proposed herein to provide a more manageable and expandable data environment than is currently in use.

This database will ultimately be designed to contain all data from the units herein referenced. It must support the standard queries of interest to these units; it must support the data needs of their application programs; it must provide the reporting capabilities they need; and, it must support the data entry and management functionality required. Furthermore, it must provide seamless integration with GIS.

From this point onward, the database will be referred to as the ***universe database***. The database is unified in that it conceptually combines/keeps the current local databases managed within each individual branch. The database is relational in its structure. Its implementation tool is ***ORACLE***. The purpose in developing the universe database is to make data entry more efficient, data query more manageable, and effective data exchange a reality among the stakeholder units.

1.3 EXISTING DATABASES

There are presently many existing databases in use within NCDOT by the stakeholders. One activity undertaken was to review the scope, nature, content, field definitions, and use of the existing databases and the attributes they contain, to gain an understanding of the data resource needs as currently embodied in existing data files. This was critical so that the design the new relational DBMS meets the existing needs of the stakeholders and accommodates all of their data. The following are a comprehensive listing of databases currently maintained by NCDOT by the stakeholders

1. ***Traffic Surveys Unit*** – Data Provider

	Database Name	**Format/Source**
a)	Annual Average Daily Traffic Data (AADT)	Mainframe
b)	Special ADT Counts	Access
c)	Turning Movements	Access
d)	Vehicle Classifications	Access
e)	ATR Data	ASCII

2. ***Traffic Planning Unit*** – Data User

   Not presently generating any databases. Conducts long-range thoroughfare studies for cities and towns incorporating Annual Average Daily Traffic (AADT) projections onto local highway networks. The unit needs access to various databases.

3. ***Traffic Forecasting Unit*** – Data User

   Not generating any databases. Develops reports that contain future traffic projections for existing and proposed roads. The unit needs access to various databases.

4. ***GIS Unit*** – Data Provider

	Database Name	**Format/Source**
a)	Mileage Inventory (Universe File)	Mainframe
b)	Street Name Inventory (Secondary Roads)	Mainframe
c)	Location Inventory (Feature File)	Mainframe
d)	Supplementary Mileage Inventory	Mainframe
e)	Various GIS Data	ARC/INFO

5. ***Traffic Safety Unit*** of ***Traffic Engineering*** – Data Provider/User

	Database Name	**Format/Source**
a)	Crash Database (w/DMV)	Oracle
b)	Street Name Directory	Oracle
c)	Ordinance Database	Mainframe

6. Other Units ***Traffic Engineering*** – Data Provider

	Database Name	**Format/Source**
a)	Sign Inventory	N/A
b)	Signal Inventory	Mainframe
c)	Railroad Crossing	Oracle
d)	Traffic Control	N/A

7. ***Pavement Management Unit*** – Data Provider

	Database Name	**Format/Source**
a)	Pavement Inventory	Oracle

8. ***Other Data Sources*** – These databases are not treated in this report. However, they are important databases and should be addressed in any future work.

	Database Name	**Format/Source**
a)	Bridge Inventory	Mainframe
b)	Functional Classification	Mainframe
c)	Paving Priority	Mainframe
d)	Highway Expenditures	Mainframe

1.4 DATABASE DESIGN CONSIDERATIONS

The primary focus of this report is on database design for the universe data. The main goal is to develop a structure for the relational database itself, i.e., to determine its content and organization. At the same time, consideration is given to a number of other design considerations, as enumerated below, which are broadly applicable to all stakeholder's data and which are being considered in the comprehensive database design.

The items enumerated below represent some of the most critical issues and needs facing state DOT's related to enterprise-wide data and information representation, management, and use [Hall 91, Wang 95, Wang 98]. It should be noted that no attempt is being made to suggest new fundamental information system frameworks or approaches. Rather we seek to identify those information system issues that are present in transportation data that significantly complicate the information system design, development, management, and use process. Upon identification and study, appropriate ways to deal with these issues in system design emerge.

It should be noted that the spatial component of data certainly is the unique and critical integrator for transportation. At the same time, special information management design considerations also emerge from the nature of state DOTs themselves, and are influenced by both their size and historical development as well as the size and historical development of their databases.

1.4.1 Spatial Data

The key to data resource integration is the fundamental way in which information is referenced to the physical world spatial data; i.e., data is joined together by a geographic reference of some sort. Residences, for example, are located using a street address, highways are located using posted route numbers, and items along a roadway may be located by specifying a distance from some landmark or otherwise recognizable starting point. The common thread among all these references is location. Yet location is described in vastly different ways and the precision and accuracy of the description varies widely as well.

Location description must be sorted out to achieve a true integration among transportation information resources and to more fully and effectively use advanced information technology tools to increase institutional productivity and effectiveness in managing transportation infrastructure. Location data must be accurately and universally represented in the spatial component of the framework of the database. Only then can true integration be achieved. Doing so will bear significant fruits in terms of facilitating the use of well understood data relationships and the identification of previously unknown (or simply unavailable) relationships; both between data and between DOT organizational units.

New interest is generated in other unit's data when that data is spatially referenced in such a way that one can spatially access it. Thus, there is the possibility of having new data *users* as well as new data *uses*. Consider, for example, the common need to issue permits for routing oversize or overweight vehicles. This activity requires information about pavement design, bridge design, load ratings, lateral clearances, environmental sensitivity, etc. Clearly such data must be obtained from a diverse collection of possibly distributed organizational units and their databases.

Consider also a second example that illustrates the need to incorporate new data. Two new applications that have previously not been broadly supported (but are clearly useful) are worth mentioning. These are billboard and sign inventories and locations. Both have a variety of useful attribute data to be stored and both need to utilize the common spatial representation shared by other applications. In doing so, interesting new relationships between signs and billboards on one hand, and any other attribute(s) of the roadway network, on the other hand, can be determined. We may thus, for example, investigate the relationship between sign locations and traffic accident locations to study possible correlations. Without a common and standardized spatial representation such linkages cannot be made without manual intervention, if at all. Thus, the criticality of spatial data and its accurate representation cannot be over emphasized.

1.4.2 GIS Interface

The unified relational database should support direct and seamless integration with the GIS so that data entry, searching, and queries can be issued spatially via the GIS directly to the relational database. That is, the database should support traditional GIS spatial queries and analysis in addition to the broad range of data analysis capabilities normally provided by a relational DBMS.

With the ready availability of ever more powerful GIS software it is critical that DOTs make use of it to perform complex spatial analysis. However, to do so requires the availability of that data. In a carefully designed transportation information system the database will provide it. But for that data to be effectively used requires a seamless coupling with the GIS. The DBMS and GIS must work hand in hand. Each must be used when it's particular strengths are required. Yet to do so requires a seamless interface between them.

1.4.3 Spatial Database Interface

Any linear spatial data access or capabilities currently in use by the stakeholders must be able to be performed directly on the relational database itself (through the use of the DBMS) in addition to being able to be performed by the GIS. These include, in particular, linear spatial search and spatial queries via the DBMS. That is, when GIS is not available, spatial analysis should still be able to be performed using the DBMS alone.

Linear queries using a GIS are well understood. Issuing linear spatial queries to traditional transportation databases, however, is a rather novel concept. A multitude of legacy databases presently exist. Enabling those databases to support spatial queries significantly enhances their value, thus significantly enhancing the utility of data already in place at most DOTs. Thus, a limited spatial analysis capability can be achieved without the wide distribution of costly GIS software. The value of existing data is significantly enhanced.

1.4.4 Existing Application Programs

This report takes into consideration the existing computer application programs used by each of the stakeholders. One major objective of this work is to provide for the data needs of each stakeholder. It is essential to identify the scope, nature, content, field conditions, and use of the required input and output of the existing software application programs currently in use to effectively support their data needs. All data items that are required to support these application programs must be included in the database schema. This involves a comprehensive analysis of all of the uses (applications) of data. It involves a complete understanding of the technical business conducted by each unit and mandates that such business be supported. Additionally, consideration must be given to new applications being planned, designed, or developed.

1.4.5 Future Data Needs

Not only should the new database design take into consideration existing applications, but it should also take into consideration projected data needs to support new applications. At the same time, future data needs are often anticipated rather than concrete. The database design, therefore, should consider them, but should fundamentally be grounded in and meet the data needs of current applications.

1.4.6 GPS

This report considers the impact of GPS and its incorporation into the proposed LRS. GPS can be used in one of two ways: (1) to determine the location of a given object, or (2) to find an object given a location. The former is measurement (used in surveying) and the latter is navigation (used in ships, cars, and cargo trucks). Today, more and more organizations are using GPS for collecting location data. As such, it must be accounted for in the database design.

The proposed LRS uses unique road identifiers (LRS-ID's) and mileposts to locate linear attributes. That is, it determines location by measurement along a line from a starting point regardless of the path that the line takes. GPS, on the other hand, determines location as a precise position in space and its measurement system is a coordinate system.

GPS coordinates are not mileposts or linear measures. However, GPS can be incorporated into the universe database by providing accurate coordinate positions for nodal entities such as intersections and county boundaries. These coordinate positions can be converted to mileposts through the application of filter programs (See Section 2.3). Or, they can simply be stored as coordinates (as attributes in the database) and programs can be written to use them as needed by a particular organizational unit. See Section 2.2.3 and Table 14 for an example.

One possible organizational dilemma is that GPS can be more accurate than digitized GIS maps. This could result in the incompatibility of location data. That is, an existing point, as stored in GIS, may be positioned differently than the GPS coordinates of that point. Put another way, the actual positional data that was field measured with highly accurate instruments and stored in the database can be more accurate than the positional data of the line work that was digitized into the CAD or GIS systems. For example, consider the GIS lines representing our most accurate positioning of a highway. A highly accurate GPS point measurement of the highway centerline might yield a location that does not even land closely on the GIS line. If not, how is the GIS line-work adjusted? This is a future issue that must be resolved as GPS and other positional measurement instruments become more widely used.

1.4.7 Positional Accuracy

Current GISs and DBMSs do not directly provide information on the quality of the data they present to the user. This shortcoming is becoming increasingly recognized as critical. Leica Geosystems AG, for example, is working with ESRI to design core GIS technology, including data structures and software tools to operate on those data, to service the unique needs of survey data and land record data. Such efforts are needed to enable DOTs to bring to bear the full power of DBMSs and GISs on transportation data.

The previous discussion regarding GPS highlights the issue of positional accuracy and precision. Accuracy is the relationship between the value of a measurement and the "true" value of the dimension being measured, that is, the correctness of the result. Precision describes the degree of refinement with which the measurement is made, for example, to additional decimal places. On one level, we are concerned with the accuracy and precision of reported data. First was the GPS receiver accurately positioned (accuracy), and second, what is the range of variability in its reading (precision)?

It is critical to know, for each field data gathering method or procedure, the degree of accuracy that can be obtained by those using the equipment. It is also critical to know the precision of the instrument, procedure, or method. This information needs to be known and used in downstream applications. It must be reported to users as well. Otherwise, users tend to assume that the value given is the actual value. But if the value is actually ten feet from where the number tells the user it is, the user needs to know and account for that.

Positional data is a key ingredient in transportation information management systems. Factors that contribute to the existence of positional data errors include the sources of positional data; the nature and volume of these data; the methods for acquiring, storing, and modeling them; the processes they can undergo; and the ways in which they can be presented. In turn, these errors affect the reliability of the information derived in and by applications. Tools must be made available to enable the user to fully understand the accuracy and precision of positional data.

Finally, the accuracy needs of the user must be considered so that gross mismatches between what a user receives and what they need in the way of positional accuracy do not occur. Related to this are the differing accuracy requirements between organizational units, which can also result in mismatches. Traffic Survey units, for example, require less precision and accuracy than Pavement Management Units. Accident location reporting requires a high degree of accuracy and precision. This is an issue that must be carefully studied and documented to ensure that information system design reflects accurately the organizations needs.

1.4.8 Data Maintenance and Management Tools

A central and essential activity is the maintenance and management of the Statewide Planning database. In a later activity the tools required to do so will be designed and developed. This report was developed keeping in mind the overall requisite maintenance and management functions so that the resulting relational database design efficiently supports those functions. A consistent set of procedures and methods must be developed to support this activity. Along with these organizational implementation tools must come a user-friendly interface that supports efficient identification and location of missing and incorrect data, respectively [Sadek 96].

The tools needed to maintain and manage the database are those which comprise the traditional scope of data language functionality.

1. Data definition (DDL). Extensions or changes to the structure of the database.
2. Data manipulation (DML). Changing, inserting, or deleting individual data item values.
3. Data query (DQL). Ad-hoc queries issued against the database.

There needs to be a clear identification of management *responsibility* for different data items. For example, the data representing the base roadway spatial network, which is used organization-wide, has such an impact that great care must be taken to ensure its integrity and utility to all users. Therefore, a clear assignment of the responsibility for this aspect of the database must be made and must be made known throughout the organization. See Section 1.5 for additional discussion of this topic.

1.4.9 Data Acquisition and Reporting

Appropriate consideration must also be given to the overall requisite data acquisition and reporting functions so that the resulting design efficiently supports those functions. The previous section related to maintaining the database. This section discusses putting things into the database (data acquisition) and getting things out of the database (reporting).

Data acquisition must account for the emergence of new technologies (e.g. GPS) which may alter the locational precision of the data and which might be significantly influenced by field methods, procedures, and training requirements. Additionally, current data collection methods, procedures, forms, etc. must be “matched” to the corresponding data items in the database. That is, field procedures must be effectively accommodated in the database. Furthermore, the FHWA Highway Performance Monitoring System (HPMS) reporting requirements must be met via the specified database design. Any other standard reports must also be able to be generated and the databases must accommodate the creation of new reports as well (and it will).

1.4.10 Data Migration

Consideration must be given to the migration of existing databases and data files to the newly structured and conceptually unified database. For each identified database and data item therein, a new location for that data needs to be specified and illustrated. That is, it must be shown how and where each stakeholder’s data is stored and how the stakeholder gains access to it. This alludes to the complexity of the whole information system problem. Not only must a new schema be investigated, designed, and developed, but a major effort of transforming all legacy data resources into the new schema must occur. The magnitude of this effort can be enormous.

1.4.11 Regeneration of Existing Database

Existing applications are tightly coupled to existing data files. During the actual development of the new database and for some time after its implementation, the application programs will require access to data in its original format. This need will persist until those application interfaces are rewritten to accommodate the new data format. Until that time, it will be necessary to regenerate the existing data files from the new database schema so that existing applications can continue to run. This is essentially a transition cost, i.e., it would not need to be borne if a new system were being completely developed from scratch but is essential for a system being transformed.

The most obvious example of this situation is the HPMS report. Historically, this report has been generated using the universe file and a mainframe application. If no other changes took place, we would have to be able to reconstitute the current universe file from the new universe database so that the existing HPMS application could run off of that file and generate the HPMS reporting data.

However, under development is a computer program to generate this report from an Oracle universe "table" instead of from the universe file. This is a single multi-attribute Oracle relational database table with exactly the same columns and content of the universe "file." Thus, the new universe database schema will have to be able to reconstitute this universe relational table so that future HPMS reporting requirements can be met. At some time in the future, the HPMS reporting application can be rewritten to run directly from the new universe database schema. Until that time we will need to be able to "regenerate existing databases."

1.4.12 Data Dictionary

As part of the review of existing databases, applications, and files, and new data needs, a comprehensive data dictionary needs to be compiled. This data dictionary should include at least three major categorizations of data: spatial, tabular, and temporal data.

Spatial data provides information about topology and geometry. Topology describes the connectivity between components or elements of the system. For example, a section of roadway may be said to be bounded by two intersections. Geometry describes the precise location in space of a component or element. For example, an intersection may have an NEZ state plane grid coordinate location.

Tabular data describes the individual attributes or characteristics of a component or element. A roadway will have some width and number of lanes, for example. An intersection will have attributes to describe the signaling. Finally, temporal data describes, in some way, the relationship that components or elements have with time.

The data dictionary should provide complete definitions for each data item, identify both the organizational unit(s) and application(s) using (responsible for) that data item, data type, data domain and range, and provide either typical or comprehensive value sets for the data item. The data dictionary should be organized and categorized into groupings related to stakeholder and applications needs for locally used data. It should support data dictionary analysis and standard queries.

1.4.13 Data Dictionary Analysis

One purpose of a critical analysis of all data items is to identify those data items that could be omitted from the new database. A second purpose is to resolve conflicts, formalize naming conventions, and standardize definitions, thus obtaining a more concise, robust set of data items to meet all previously determined needs. A third purpose is to appropriately assign responsibility for, and ownership of, data, i.e., to ensure that it is properly positioned in the organization.

Three conditions contribute to the need for a critical analysis of the new comprehensive data dictionary. First, it is anticipated that over time a mismatch has occurred between the supply of data on one hand, and the anticipated program demand for, or use of data, on the other hand. It is expected that some existing data items may no longer be needed. These items should be identified and removed. Maintaining unnecessary data is extremely costly and is to be avoided.

Second, it is known that secondary data items have been created from combinations of other more fundamental (or base) data, thus proliferating a degree of redundancy of existing data. Furthermore, different ways of referring to and using data also result in redundant data items. These data items should be identified and removed if not needed in recreating data sets in previous formats.

Finally, it is known that various organizational units maintain data for other units unnecessarily. That is, some units could assume ownership of, and responsibility for, data items within their domain or sphere of activities. Regardless of the original reasons for data maintenance and management by any unit, all items should be examined to determine their rightful or most logical home. For example, Section 4.6 (Existing Databases) shows the GIS Unit maintaining data that is of primary interest to Pavement Management (Number of Travel Lanes, Surface Width, etc.), Traffic Surveys (Design Hour Speed, PTC Count Station, etc.) and Forecasting (Future AADT, Future AADT Year). Both Pavement Management and Traffic Surveys possess the expertise, knowledge, interest, and ability to assume control of these data items. Traffic Forecasting, however, may not currently possess all of the requisite resources to maintain their data and, therefore, it may be more appropriately maintained by Roadway Inventory.

1.4.14 Temporal Data

Time comes into play in many situations and must be considered and dealt with in the information management system design. One of the unique aspects of a State DOT database design is that the physical infrastructure can change over time; new roadways can be built, others can be abandoned or moved. These infrastructure changes are time-critical spatial network changes and they impart unique situations on the database design.

Consider accident locations, for example. Accident data may have been collected for many years at a particular intersection. If that intersection was later relocated problems can arise in interpreting accident location in the vicinity or region. A discontinuity in time and location can have an adverse effect on historical analysis of data that has fundamentally changed.

Time thus influences the interpretation of historical data. A change in roadway width from two to four lanes is a significant change in the physical infrastructure. When that change occurs is important because it influences how we interpret other data. Traffic survey counts, for example, would significantly change in such a scenario. But if there isn’t a way to establish a correlation between the roadway widening activity and the counts, this change in count data could cause concern. Temporal data, therefore, is a consideration in transportation database design. It is thus discussed in greater detail in Sections 3.2 to 3.5.

1.4.15 Data Exchange

A complete understanding of the scope and extent of data sharing is an important consideration in database design as well. Historically many State DOT organizational units have operated in isolation and data sharing was at a minimum. However, the proliferation of standardized software tools, the need to access data owned or maintained by others, and external demands for data all mandate that more and more data will be shared.

Individual users within an organizational unit are operating in a much more automated office working environment. This environment itself promotes the desire to enhace data sharing within that unit. Increasingly complex design situations are mandating cross unit data sharing. Relationships between State and Federal agencies mandate data sharing; the most prominent example being the HPMS data reporting requirements. A critical need may also exist, for example, between a State Highway Department and its Division of Motor Vehicles that would significantly benefit by the ability to readily share and analyze data.

Finally, the general public is developing a larger and larger appetite for a variety of data. As transportation significantly impacts our environment and other aspects of our infrastructure increasing demands will be made to share transportation data. Consideration, thus must be given in the transportation information management system design, to how much sharing will occur, with whom, how often and in what form. The delivery vehicle(s) for that sharing must be considered as well.

1.4.16 Expanding the Organizational Scope

How is a DOT to proceed with an information system development effort? The successful experience at NCDOT began with a core stakeholder group. That group was of sufficient size to involve enough participants to get the work done and encompassed a wide enough variety of data to support a generalized and expandable solution. Other new organizational units can then be incrementally included in such an information system development effort if they adopt the methods and procedures outlined in the original information system design. Other interested units and branches may include Construction, Maintenance, Permitting, Bridge Management, Location and Surveys, and others.

1.4.17 Training

Because of their large size, training is an issue to be seriously considered by state DOTs. The adoption of any new methods, procedures, or software requires training consideration. The adoption of a relational database format, for example, will necessitate some employee training in this method. The adoption of a specific relational database tool will also necessitate employee training to use that software. The amount of this training will largely be determined by the level of standardization that can be achieved and by the organizational structure. For NCDOT we found that standardization on a base linear referencing method, and on the relational database format, provided an ideal level of standardization that could readily be supported via training.

1.4.18 Procedures

The database design will, as any does, be based on assumptions particular to the domain of use of the database. Special rules or circumstances dictate the way the database is designed, e.g., there may be assumptions regarding name changes of secondary roads as they cross county boundaries. These need to be identified, organized, and incorporated into a rules and procedures manual for NCDOT database designers. The goal is to develop a guideline document that enables the developers of new data sets to create databases and data sets that are compliant with the new LRS.

1.5 OWNERSHIP AND RESPONSIBILITY

A key emerging concern regarding data and information in state DOTs is that of ownership and responsibility. Each unit's daily operation depends, in part, on the data it generates, uses, shares, and maintains. Therefore the security and accuracy of that data is of paramount concern to the "owning" unit. This organization's impact must thus be accounted for in any unified information management system design and development effort.

The recommended information system design encompasses a conceptually centralized database but with distributed ownership and responsibility. That is, the nature of the relational database model is such that free exchange of relational database tables is easily achieved. This assumes, of course, a common relational database tool and the adoption of a common base LRM. Given that these two requirements are met, data exchange will be immensely facilitated as a result of adopting the unified database design. This may be the one most significant results of this investigation – global access, yet local ownership and responsibility. Furthermore a suite of commercial database software products running on servers provide the functionality to implement such a design.

At the same time, it must be recognized that not everyone needs instant access to all data. The nature of the majority of the data is such that one unit generally works with its data extensively and holds significant organizational responsibility for it. Usually the needs of others for that same data are less immediate and less frequent. Some data may even be proprietary or protected from access by others outside the organizational unit. Most data, therefore, does not need to reside on a central server and be continuously available to everyone. Rather, it can usually reside with its host organizational unit and be made readily available upon request.

Clearly, some data will need to be shared more quickly and more often than other data. The point here is that there is also much data that does not need to be shared so quickly and often. This data can remain under more local control. The fact that it is in a relational format, that it is on ORACLE, and that it is linked to the standard LRS means that it is available and can readily be shared. At the same time, the fact that it exists in a client/server environment means that there are effective ways to limit access to it if necessary.

1.6 FORMAT

This report presents a relational database schema for the universe data. The schema in each part is designed to work with the data as it currently exits and it incorporates the proposed linear referencing system.

The format of the proposed design is presented in two forms. One set of tables are provided by identifying the table name along with the primary key(s) and attributes as follows:

TABLE NAME (**Primary Key**, Attribute 1, Attribute 2, Attribute 3, …, Attribute n)

Where appropriate, actual tables with representative data are presented. These tables are intended merely as a guide for illustration, where necessary. They are provided in the following form:

Table # – TABLENAME

Primary key	**Attribute 1**	**…**	**Attribute n**

2.0 SPATIAL TOPOLOGY

2.1 THE CONCEPTUAL SPATIAL TOPOLOGY

This section presents the most fundamental conceptual schema for implementing the spatial topology of the recommended base LRS. A centralized approach is recommended for NCDOT, but a distributed approach is also presented so that the reader can better understand the concepts and see how they represent essentially the same solution.

2.1.1 Distributed Spatial Data

In the universe database, linear elements, such as pavement type and speed limits, are spatially defined using the base LRS. This requires a unique ***LRS ID*** for each road in the route system and a ***MP1*** and ***MP2*** to mark the beginning and ending measurements along the specified road. Table 1 shows the general table structure for linear elements. Point features, such as railroad crossings, will be spatially defined with the unique LRS ID and a ***Milepost*** measurement along that route as shown in Table 2.

Table 1 – LINEAR TOPOLOGY TABLE

LRS ID	MP1	MP2	Any Linear Attributes
515	0.00	1.00	X
623	68.00	69.40	X
623	69.40	71.50	X

Table 2 – POINT TOPOLOGY TABLE

LRS ID	Milepost	Any Point Attributes
623	69.20	X
624	106.20	X

Table 3 – INTERSECTIONS

LRS ID	Milepost	Node ID
643	32.2	N25
643	38.9	N89
745	4.7	N25
745	7.8	N89

Table 3 represents an example of the spatial topology of intersections in the LRS route system. Anywhere an LRS route crosses another LRS route an intersection is defined. The easiest way in which to represent an intersection is through a ***Node*** identifier. The **Milepost** field is used to represent the distance along the LRS route where the intersection is positioned. Each intersection must be represented by at least two records containing the same ***Node ID***. This is because there are at least two LRS routes required to form an intersection and the node will be located at a different milepost distance for each.

Nodes are not the only way in which intersections can be represented in the topology. However, they do present the database with an easier query option. Information related to each intersection could be queried using the **Node ID** as the primary key. Nodes are separately identified because they are an integral part of the network topology. Table 3 is included here only to illustrate the concept of nodes and their relationship to the LRS system. In Section 2.3 we will see necessary extensions to the INTERSECTIONS concept.

Other crossing features, such as railroad crossings and county boundaries, do not need to be assigned nodes because they can be represented solely on the basis of the **Milepost** measure. That is, an intersection is composed of two or more intersecting roads, where a railroad crossing has only one intersecting road. Therefore, attributes can be assigned to a railroad crossing with only the **LRS ID** and the **Milepost** fields as the composite key.

It should be emphasized that each unit may choose to continue using a different location method in their data collection and reporting. For example, The Traffic Safety Systems Unit describes accident locations based on a distance from a particular intersection along a roadway (anchor point reference). They currently use parameters such as POSTED ROUTE, INTERSECTION, and DISTANCE to locate accidents. If a unit does not adopt the proposed LRS in its data collection and reporting activities, then a conversion mechanism must be developed to convert a particular referencing system into the proposed LRS. These conversion mechanisms are often referred to as filters. These are further discussed in Section 2.3.

However, for the system to truly work at its intended optimum with regard to data collection, maintenance, and distribution, all stakeholders should adopt the chosen common LRS. Otherwise, the current incompatibilities will remain and the goals noted above will not be fully achieved.

Tables 1, 2, and 3 are intended to conceptually demonstrate how the spatial topology works. In actuality there will be an assortment of non-spatial attributes linked to these spatial keys. However, the database will not always use the **LRS ID** and **Mileposts** as keys. In some instances, other keys will be required, such as “Station Number.” In such a case, the **LRS ID** and **Milepost** fields will serve as attributes linked to the primary key.

2.1.2 Centralized Spatial Data

In the distributed approach all **LRS IDs** and **MP1** and **MP2s** are explicitly stored in each and every data table. This presents a serious problems if changes need to be made in the values of any of these attributes. Such changes will especially occur if values for mileages in the underlying network are changed. This is very likely in the event of a new field roadway inventory.

Consider a scenario using Table 3. Assume that the Roadway Inventory Section determines that the milepost for **LRS ID** N89 has been incorrectly entered as 38.9 and its value should really be 37.9. Also assume that Traffic Surveys has used this base network data table to build its AADT table as follows.

AADT

LRS ID	MP1	MP2	AADT
643	32.2	38.9	2164

Clearly the value of 38.9 must be changed in the AADT table (and many others for that matter) as well as in the INTERSECTIONS table. This presents a significant maintenance problem for any Sections or Units which link their data in any way to the base network. It is for this reason that the most accurate base distances be established early and entered into the system in its first operational capacity.

However, recognizing that changes in the network will continue to occur as technology improves both measurement accuracy and the ability of the unit to gather more accurate data; and that new additions to the network are an inevitability; an alternative implementation was sought to mitigate the obvious maintenance problem. Note that this problem is unique to linear roadway elements; point events are measured independently.

The centralized spatial data approach requires a unique **Place ID** to identify each and every linear segment of interest to any database user. **MP1** and **MP2** are used to mark the beginning and ending measurements, thus identifying the location of the segment along the specified road. Table 4 shows the general table structure for places.

Table 4 – LINEAR TOPOLOGY TABLE

LRS ID	MP1	MP2	Place ID
515	0.00	1.00	1
623	68.00	69.40	2
623	69.40	71.50	3

Using this table one can determine the location (LRSID, MP1, MP2) given a **PlaceID**, or one can determine the **PlaceID** given a specific location. Attribute values are then stored as shown in Table 5.

Table 5 – ATTRIBUTE TABLE

PlaceID	Any Linear Attribute
1	X
2	X
3	X
4	X

The POINT TOPOLOGY and INTERSECTION tables remain the same and conceptually require no changes.

The key difference between the centralized spatial data approach illustrated in this section and the distributed spatial database approach illustrated in the last section is that in the centralized approach all spatial data is centralized in one table (LINEAR TOPOLOGY TABLE). As a result, changes are all localized to that table alone. Maintenance is thus made easier from the point of view of all users. But this does not mean that maintenance becomes a trivial exercise. Rather, it is localized in such a way that a far greater assurance can be obtained that changes will correctly be incorporated into the database and all users can have confidence in the validity and integrity of the data.

2.2 GENERIC EXAMPLE: IMPLEMENTATION INCLUDING TOPOLOGY AND ATTRIBUTES

This section presents an example of how database tables are constructed using the proposed spatial topology. Figure 1 in the Appendix represents a sample route system with posted routes and intersections shown. Nodes are used to represent intersections and county boundaries. Figure 1 shows these intersection and county nodes. Intersection nodes are labeled N(n), while county nodes are labeled C(n). This is for demonstration purposes only and has no significance. Tables 6, 7, 8, and 9 demonstrate a generic representation of how the distributed tables may be laid out in the ORACLE database.

2.2.1 Distributed Spatial Data

Table 6 stores the names of posted routes that are associated with a particular LRS route. There may be one or more posted routes associated with a given LRS route. **MP1** and **MP2** fields are used to record the LRS measure along the LRS route for which the specified posted route designation applies.

Table 6 – POSTED ROUTES

LRS ID	MP1	MP2	Posted Route
515	0.00	1.00	NC 115
623	68.00	71.50	NC 78
624	105.00	108.50	NC 54
742	25.00	26.80	NC 115
742	25.00	27.25	NC 65

Table 7 stores the nodes that represent an intersection. These nodes may be queried to reveal information related to each intersection. Note that in Figure 2 in the Appendix, nodes have been placed at the ends of each route as they are drawn. These routes are actually continuous even though nodes are shown. This is only for demonstration purposes and will not occur in the actual database representation except for roads crossing the boundaries of the State.

Table 7 – INTERSECTIONS

LRS ID	Milepost	Node ID
624	106.90	N4
623	69.90	N5
742	25.70	N4
742	26.45	N5
742	26.80	N6
515	0.00	N6

Table 8 stores the nodes that represent county boundaries. These nodes may be queried to display information such as county names.

Table 8 – COUNTY BOUNDARIES

LRS ID	Milepost	Node ID
623	68.55	C2
624	105.45	C1

Table 9 catalogs the locations of all railroad crossings in the sample route-system. The Milepost field is used to reference this point feature. Refer again to Figure 2 (Appendix). Any number of additional attribute fields could be added to this table or a unique identifier could be added for each crossing.

Table 9 – RAILROAD CROSSINGS

LRS ID	Milepost
623	69.20
624	106.20

2.2.2 Centralized Spatial Data

Tables 10 and 11 correspond to the centralized implementation of the information previously represented in Table 6. Table 11, however, is extended to contain two additional fields – **Posted MP1** and **Posted MP2**. These fields provide equivalent posted mile markers to the internal LRS measures and enable the query of posted measures for certain reporting purposes. These fields provide the recognition that field milepost measures are different from posted measures and, in addition, they document the relationship between the two.

Table 10 – PLACES

LRS ID	MP1	MP2	Place ID
515	0.00	1.00	1
623	68.00	71.50	2
624	105.00	108.50	3
742	25.00	26.80	4
742	25.00	27.25	5

Table 11 – POSTED ROUTES

Place ID	Posted Route ID	Posted MP1	County 1	Posted MP2	County 2
1	NC 115	14.30	X	15.30	X
2	NC 78	42.90	Y	2.95	X
3	NC 54	36.80	Y	3.05	X
4	NC 115	12.50	X	14.30	X
5	NC 65	12.50	X	14.75	X

The data in the previous INTERSECTIONS (Table 7), COUNTY BONDARIES (Table 8), and RAILROAD CROSSINGS (Table 9) tables remains the same. One additional table that will be needed is the COUNTY table. This table establishes a relationship between each county and the places in that county. Thus, all routes within any county can be easily determined. This table would list *all* places that are fully contained within a county and *no* places that cross county boundaries would be included. This table, then, implements the concept of a county as being an attribute of the physical roadway, as it should be.

Table 12 – COUNTIES

Place	County
1	X
4	X
5	X

2.2.3 Distributed Non-Spatial Attribute Data

The previous tables stored information fully describing the topology and geometry of the route system. Now we illustrate the storage of attribute information on this network.

Table 13 stores the pavement condition found along each route in the sample route-system for the distributed approach. Pavement condition is a linear attribute so it uses an **MP1** and an **MP2** as a spatial delineation. Refer to Figure 3 (Appendix) for a pavement condition map representation using the same sample route system as in Figure 1 (Appendix).

Table 13 – PAVEMENT CONDITION

LRS ID	MP1	MP2	Pavement Condition
515	0.00	1.00	Good
623	68.00	69.40	Moderate
623	69.40	71.50	Good
624	105.00	106.70	Good
624	106.70	108.50	Excellent
742	25.00	26.45	Poor
742	26.45	26.80	Moderate
742	26.80	27.25	Poor

Finally, one additional attribute table is illustrated. This table deals with GPS data; it identifies those points within the overall highway network whose precise geometric location is known. The form of this data is shown below in Table 14.

Table 14 – NODE COORDINATES

Node ID	N	E	Z
N1	121	412	312
N2	115	513	336
N3	201	378	420
N4	203	459	378
N5	207	488	245
N6	209	569	365
N7	214	642	398
N8	298	449	425
N9	299	502	403

N10	296	523	397
C1	136	414	357
C2	145	516	437

In this table, a subset of LRS nodes would be represented; those whose precise North Carolina grid coordinate positions are known. This table establishes a link between the geometry of the lines in the stored network (by such tools as ARC/INFO and Microstation) and their actual location within the state. These coordinate values may also establish a link with GPS data collection devices, as these are increasingly used within NCDOT and elsewhere.

2.2.4 Centralized Non-Spatial Attribute Data

The equivalent centralized schema implementation is as follows.

Table 15 – PLACES

LRS ID	MP1	MP2	Place
515	0.00	1.00	1
623	68.00	71.50	2
624	105.00	108.50	3
742	25.00	26.80	4
742	25.00	27.25	5
623	68.00	69.40	6
623	69.40	71.50	7
624	105.00	106.70	8
624	106.70	108.50	9
742	25.00	26.45	10
742	26.45	26.80	11
742	26.80	27.25	12

In Table 15 the first 5 **Places** are those which are used to identify posted routes (attribute Table 11). **Places** 6-12 identify roadway segments with a certain pavement condition (attribute Table 13). Note that **Place 1** is used by both attributes. That is, there is a segment of roadway that has two attribute values over a common portion of its length. Table 16, below, identifies those pavement conditions.

Table 16 – PAVEMENT CONDITION

PlaceID	Pavement Condition
1	Good
6	Moderate
7	Good
8	Good
9	Excellent
10	Poor
11	Moderate
12	Poor

There would be no change in the ANCHOR POINTS (Table 14) table.

2.3 LRS FILTER PROGRAMS

A collection of LRS filter conversion programs will be needed to transform each individual unit's referencing system into the proposed LRS. These filter programs will be customized to the type of referencing system that is currently used by a particular unit. In some instances, a further set of conversion programs will be required by a unit to format data into some other standardized format. For example, the GIS unit produces an annual HPMS report using a specified format, which is currently based on the Universe and Supplementary Files. A conversion program must be written to transform their data, as it will exist in the new universe database, into the HPMS format. This was previously discussed in Section 1.4.10.

Filter programs will need to be highly optimized algorithms of modular code, much like the LRS Join algorithm discussed in Section 2.4 below. The filter programs that have been identified and are in use by Stakeholders include link node, posted route, and anchor point reference. Others, including address geocoding and coordinates (See Section 1.4.5 and 2.2.3), can also be written. Each will require thoughtful design and optimized coding.

The generalized form of the COORDINATES table was given in Section 2.2.3 (Table 14). The generalized form of the INTERSECTIONS table was given in Sections 2.1.1 (Table 3) and 2.2.1 (Tables 7 and 8). The generalized form needed for posted route filter data is given here as an "anchor point" table, that is, a modified INTERSECTIONS table.

Clearly, this is an extension of the INTERSECTIONS table previously presented. The records stored in this table provide both **LRS IDs** and **MPs** and **Posted Route IDs** and **MPs** for all spatial network intersections. No additional records need to be stored. With this information alone a filter program can be generated.

To illustrate the filter concept, assume that one specifies a roadway segment as being on NC 54 between **Posted MP** 0.75 and 1.45. Also assume that this corresponds to being on **LRS ID** 624 between **LRS MP** 106.2 and 106.9 as shown in Figure 2 in the Appendix. The appropriate entries (among others) in the INTERSECTIONS table would be as follows.

INTERSECTIONS

LRS ID	LRS MP	Node ID	PR ID	PR MP
624	105.00	N1	54	36.80
624	105.45	C1	54	0.00
624	106.90	N4	54	1.45
624	108.50	N8	54	3.05

Note that the first specified point in our example (at the RR crossing) is an event location (an attribute) rather than a node location like the second point. There can be two scenarios. Given Posted Route information, one could request LRS Route information, or the converse could also be requested.

Consider the first scenario using data from the above table and from Figures 1 and 2 in the Appendix. Given **PR MP** 1.45 on NC 54 it is easy to search the table, find the exact match, and identify **LRS MP** 106.9 on **LRS ID** 624. For point 0.75 on NC 54, however, there is no exact match so we must perform a calculation. First, we must find the two points between which the point of interest lies (0.75 lies between 0.00 and 1.45). We then find the distance from either. Our point of interest lies 0.75 miles from C1. We then find the **LRS ID** and **LRS MP** for C1 and add this value to it, giving us 105.45 + 0.75 and yielding an **LRS MP** value of 106.2. The same conversion would also work in reverse.

Additional resolution is needed here when the final database is designed. There are a number of ways to go for the formal implementation. The county must be included along with the posted route information. The county may be specified as a separate field, or, it could be included in the **PR ID** field in a manner much like it is done now in the universe file.

2.4 JOINING TABLES WITH DIFFERENT ATTRIBUTES

Once the universe database has been populated with data, users will want to query specific data items, often in combination with other data items. The database must be capable of bringing data together from different tables. Each data item will have a corresponding milepost reference. In a normal "join" operation, it is simple to bring two or more tables together using a common primary key. However, in the case of the universe database, attributes are referenced by an LRS ID and by milepost measures. It will rarely be the case when two data types can be referenced to exactly the same mileposts. Therefore, a special type of "join" operation must be performed in order to bring data types together from different tables. We call this an LRS Join.

2.4.1 LRS Join Concept

Figures 3, 4, and 5 in the Appendix provide a visual representation of different data types occupying the same road segment. Milepost measurements are used to reference each data type on an LRS route. Figure 3 shows the pavement condition at referenced mileposts for each LRS route. The data for this pavement condition scenario was previously presented in Table 13 and is reproduced here in Table 17 for reader convenience. Figure 4 shows the pavement width at referenced mileposts for each LRS route and the data for this pavement width is presented in Table 18.

There are many plausible queries that may be issued to the database with respect to pavement condition and width. For example, if a user asks "What are the roads having poor pavement condition and a width of 26 feet?" the computer must match the keywords "pavement condition" and "pavement width" with the tables containing the segmented information (Tables 17, 18).

Next, a *select* operation must be performed to identify and extract all rows from Tables 17 and 18 showing a "poor" pavement condition and a "26-ft" pavement width, respectively. The candidate rows for the query are shown highlighted in Tables 17 and 18 for reader convenience. Tables 19 and 20 are the new tables created based on the selection criteria. These tables contain all instances where there are "poor" pavement conditions and "26-ft" pavement widths.

One additional consideration is to further reduce the size of Tables 19 and 20 before combining them. For us to get an answer to our query that consideration requires that all the resulting rows have the same LRS ID. Thus, any rows in Table 19 that have an LRS ID that does not appear in Table 20 can be removed from consideration. Likewise, any rows in Table 20 that have an LRS ID that does not appear in Table 19 can also be removed from consideration. Upon inspection Table 19 remains the same, but Table 20 can be reduced to the following table as shown.

“SELECTED” PAVEMENT WIDTH

LRS ID	MP1	MP2	Pavement Width
742	25.70	27.00	26’

However, as the following discourse shows the step just outlined above is not necessary. Continuing, an overlay algorithm is used to join the tables containing the *selected* pavement condition data and the *selected* pavement width data into a new topological division where the values of interest occur together in the same segment (Figure 5, Appendix). This overlap of data is then returned to the user in the form of a combined table as shown in Table 21. Thus, the answer to the query has been provided with the assistance of the LRS.

Table 17 – PAVEMENT CONDITION

LRS ID	MP1	MP2	Pavement Condition
515	0.00	1.00	Good
623	68.00	69.40	Moderate
623	69.40	71.50	Good
624	105.00	106.70	Good
624	106.70	108.50	Excellent
742	25.00	26.45	Poor
742	26.45	26.80	Moderate
742	26.80	27.25	Poor

Table 18 – PAVEMENT WIDTH

LRS ID	MP1	MP2	Pavement Width
515	0.00	0.50	26’
515	0.50	1.00	28’
623	68.00	68.55	26’
623	68.55	69.20	28’
623	69.20	70.80	26’
623	70.80	71.50	28’
624	105.00	105.45	26’
624	105.45	106.20	28’
624	106.20	108.50	26’
742	25.00	25.70	24’
742	25.70	27.00	26’
742	27.00	27.25	28’

Table 19 – "SELECTED" PAVEMENT CONDITION (Table A)

LRS ID	MP1	MP2	Pavement Condition
742	25.00	26.45	Poor
742	26.80	27.25	Poor

Table 20 – "SELECTED" PAVEMENT WIDTH (Table B*)*

LRS ID	MP1	MP2	Pavement Width
515	0.00	0.50	26'
623	68.00	68.55	26'
623	69.20	70.80	26'
624	105.00	105.45	26'
624	106.20	108.50	26'
742	25.70	27.00	26'

Table 21 – COMBINED TABLE – LRS Join of Tables A and B

LRS ID	MP1	MP2	Pavement Condition	Pavement Width
742	25.70	26.45	Poor	26'
742	26.80	27.00	Poor	26'

2.4.2 LRS Join Algorithm

The algorithm mentioned in the previous section can be used to join Tables 19 and 20 into a combined table such as Table 21. The algorithm simply compares the mileposts for a given LRS ID from each *selected* table and returns an answer to the query based on the result of the comparisons. This section illustrates the mathematics of the LRS Join.

In order to compare the mileposts from the *selected* tables (Tables 19 and 20), a series of operations must be performed. However, the operations will only be performed where there is a match between the LRS ID's. That is, the operations will only be executed when a row from Table 20 has the same LRS ID as a row from Table 19. For demonstration purposes, Table 19 will be referred to as Table "*A*" and Table 20 will be referred to as Table "*B*." Thus, when the letter "*A*" precedes a milepost, it refers to a milepost from Table 19, and when the letter "*B*" precedes a milepost, it refers to a milepost from Table 20. For example, "*A* MP1" refers to the ***MP1*** field in Table 19.

Each of the comparisons that follow has a logical answer (True or False). The comparisons include:

1) B MP1 $<$ A MP1
2) B MP1 $>$ A MP1
3) B MP1 $\leq$ A MP1
4) B MP1 $\geq$ A MP1
5) B MP2 $<$ A MP1
6) B MP2 $>$ A MP1
7) B MP2 $\leq$ A MP1
8) B MP2 $\geq$ A MP1
9) B MP1 $<$ A MP2
10) B MP1 $>$ A MP2
11) B MP1 $\leq$ A MP2
12) B MP1 $\geq$ A MP2
13) B MP2 $<$ A MP2
14) B MP2 $>$ A MP2
15) B MP2 $\leq$ A MP2
16) B MP2 $\geq$ A MP2

Once the preceding comparisons are completed for the two rows that are being compared, the algorithm proceeds to the following set of rules. When a rule is satisfied where there is an intersection of mileposts, a row is added to the combined table (Table 21). The rules are as follows:

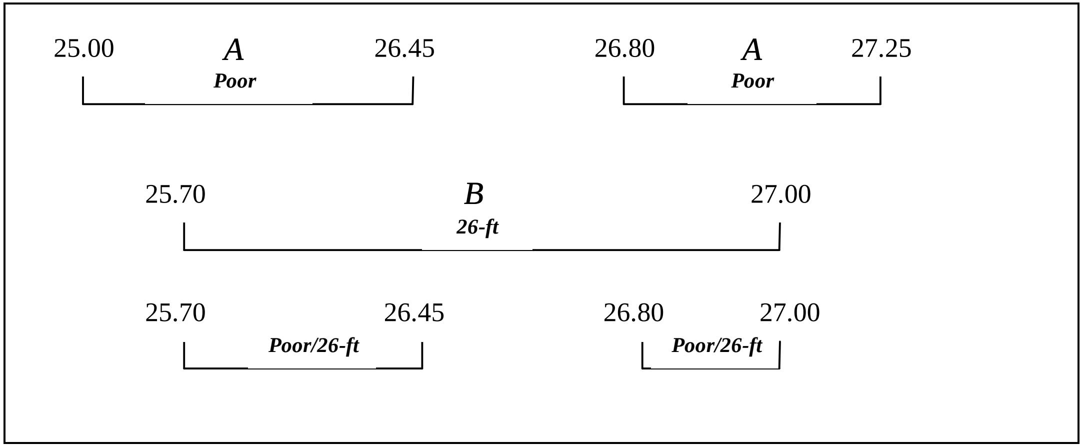
1) If B MP2 $<$ A MP1,
   there is no join.
2) If B MP1 $>$ A MP2,
   there is no join.
3) If B MP1 $\geq$ A MP1 and B MP2 $\leq$ A MP2,
   the intersection occurs from B MP1 to B MP2.
4) If A MP1 $\geq$ B MP1 and A MP2 $\leq$ B MP2,
   the intersection occurs from A MP1 to A MP2.
5) If B MP1 $\leq$ A MP1, B MP2 $\geq$ A MP1, and B MP2 $\leq$ A MP2,
   the intersection occurs from A MP1 to B MP2.
6) If B MP1 $\geq$ A MP1, B MP1 $\leq$ A MP2, and B MP2 $\geq$ A MP2,
   intersection occurs from B MP1 to A MP2.

In English, the following interpretations are associated with these conditions.

1) The segments do not intersect and segment *B* occurs before segment *A*.
2) The segments do not intersect and segment *A* occurs before segment *B*.
3) Segment *B* is wholly contained within segment *A*.
4) Segment *A* is wholly contained within segment *B*.
5) *B* MP1 is positioned ahead of *A* MP1, *B* MP2 is positioned after *A* MP1, and *A* MP2 is positioned after *B* MP2. Therefore, segments *A* and *B* overlap from *A* MP1 to *B* MP2.
6) *A* MP1 is positioned ahead of *B* MP1, *B* MP1 is positioned before *A* MP2, and *B* MP2 is positioned after *A* MP2. Therefore, segments *A* and *B* overlap from *B* MP1 to *A* MP2.

One at a time, each set of mileposts from *B* is compared to each set of mileposts from *A* (having the same LRS ID) using the comparisons and rules shown above. After the first set of mileposts from *B* has been compared to each set of mileposts from *A*, the second set of mileposts from *B* is compared to each set of mileposts from *A*, and so on. After all sets of mileposts from *B* have been compared to each set of mileposts from *A*, the algorithm terminates.

Figure 7 – Intersection of Mileposts for **LRS ID** 742



2.4.3 LRS Join Example

The following example is intended to help clarify the algorithm introduced in the previous section. Figure 7 graphically shows the intersection of mileposts from Tables 19 and 20. The information provided here matches the information provided in Table 21.

In the first step of the algorithm, row 1 from *B* (Table 20) is compared to row 1 from *A* (Table 19). Since the LRS-ID's do not match, no further comparisons are made. Next, row 1 from B is compared to row 2 from A. Again, the LRS-ID's do not match so the algorithm proceeds to row 2 of B. This process proceeds until there finally is an LRS-ID match between row 6 of B and row 1 of A. Now, the logical comparisons, enumerated in the previous section, are executed. The results of the comparisons are as follows:

1) *B* MP1 < *A* MP1 - False
2) *B* MP1 > *A* MP1 - True
3) *B* MP1 ≤ *A* MP1 - False
4) *B* MP1 ≥ *A* MP1 - True
5) *B* MP2 < *A* MP1 - False
6) *B* MP2 > *A* MP1 - True
7) *B* MP2 ≤ *A* MP1 - False
8) *B* MP2 ≥ *A* MP1 - True
9) *B* MP1 < *A* MP2 - True
10) *B* MP1 > *A* MP2 - False
11) *B* MP1 ≤ *A* MP2 - True
12) *B* MP1 ≥ *A* MP2 - False
13) *B* MP2 < *A* MP2 - False
14) *B* MP2 > *A* MP2 - True
15) *B* MP2 ≤ *A* MP2 - False
16) *B* MP2 ≥ *A* MP2 - True

Next, the algorithm moves to the rules that define the possible intersections between the rows from the two tables. Based on the logical comparisons presented above, the rules are applied as follows:

1) Is *B* MP2 < *A* MP1? = ***No***.
2) Is *B* MP1 > *A* MP2? = ***No***.
   Therefore, because both rules (1) and (2) are false, a join must occur.
3) Is *B* MP1 ≥ *A* MP1 and *B* MP2 ≤ *A* MP2? = ***No***.
   Therefore, there is no *complete* intersection from *B* MP1 to *B* MP2.
4) Is *A* MP1 ≥ *B* MP1 and *A* MP2 ≤ *B* MP2? = ***No***.

Therefore, there is no ***complete*** intersection from *A* MP1 to *A* MP2.

5) Is *B* MP1 ≤ *A* MP1, *B* MP2 ≥ *A* MP1, and *B* MP2 ≤ *A* MP2? = ***No***.
   Therefore, there is no ***complete*** intersection from *A* MP1 to *B* MP2.
6) Is *B* MP1 ≥ *A* MP1, *B* MP1 ≤ *A* MP2, and *B* MP2 ≥ *A* MP2? = ***Yes***.
   Therefore, an intersection occurs from *B* MP1 to *A* MP2.

At this point, the first row of the combined table (Table 21) is obtained. Next, row 6 from *B* is compared to row 2 of *A* and the rules are applied as shown above. After row 6 from *B* is compared to row 2 of *A*, and the rules applied, there are no more rows to compare and the algorithm terminates.

This example only compared two tables. In reality, a query may involve the selection and joining of three or more tables. In that case, the preceding algorithm must be revised to compensate for the additional table(s). This will involve a more complex operation. However, it will involve the same basic steps outlined in sections 2.4.2 and 2.4.3.

2.4.4 LRS Join Assessment

Because of the complexity of the LRS Join and because it is unconventional, it is not available as a standard database operation. Rather, it must be programmed. This leads to questions regarding its speed of operation because of the large number of comparisons that must be made and the fact that every record in the file must be examined to do each comparison. Furthermore, there are on the order of up to 200,000 records in some universe files and, in the future, there could be more.

To explore the efficiency of the LRS Join tests were conducted using ORACLE to evaluate the speed of multi attribute joins. Two and three table LRS Joins were executed. Both processed the tests in seconds. As a result, it is expected that nearly all typical multi attribute queries that would need to be answered could be processed in speeds that more than meet stakeholder needs.

2.4.4.1 Distributed Topology and Geometry Tests

The experiments began by creating two tables:

SPEED_ATTRIBUTE (LRS ID, MP1, MP2, speedlimit)
SURFACE_WIDTH_ATTRIBUTE (LRS ID, MP1, MP2, surfacewidth)

These tables were created by extracting the data directly from the ORACLE universe table. (The ORACLE universe table is an exact replica of the universe file.) The universe table contains approximately 125,000 records.

The new **LRS ID** field was created as a composite of the posted route ID and the county fields of the universe table. After creating the tables, the records were compressed, that is, all contiguous roadway segments having that same attribute value were compressed into a single record. This is done because the universe table is comprised of many short links, each of which may have the same value for that attribute. If so, all of those contiguous records could be compressed into a single record correctly specifying the attribute value for the larger, composite roadway. This reduces the overall number of records in the table. In the case of the SPEED_ATTRIBUTE table the number of records was 73,635 and in the SURFACE_WIDTH_ATTRIBUTE table there were 89,041 records.

Note that, for various attributes, far fewer records are needed to characterize the roadway network with respect to that attribute than the 125,000 records in the universe file. This will enhance maintenance significantly.

Our goal in the experiment was to evaluate the speed of the LRS Join to determine its feasibility. The first query of interest was to determine which roadway segments had a surface width = 20 and had a speed limit = 55. To determine this, an algorithm was written in SQL to first select surfacewidth = 20 and place the result in a temporary table (it may be called SW = 20). The same was done for speed limit = 55 (creating a new table called SL = 55). These tables contained 29,446 and 57,563 records, respectively. They were then joined to find those records that had both values. The number of records that satisfied the query was 26,744 and the time to execute the entire process was virtually instantaneous.

At this point in the process we had satisfied a two-attribute query, on tens of thousands of records, requiring an LRS Join, in virtually no time. Clearly, ORACLE enabled us to effectively achieve our objective. Additionally, because the speed is so fast, we can create virtually a binary database. This means that most attributes will be in tables by themselves rather than with one or more other attributes. The significant gain for this is in maintenance. Binary tables, in the context of the LRS, mean the presence of an attribute and a location. Bear in mind that location is comprised of an ID and a beginning and ending MP. The SQL algorithm used for the two-attribute query was the following.

```
SELECT  a.lrs_id,
        GREATEST(a.begin_mp, b.begin_mp) as begin_mp,
        LEAST(a.end_mp, b.end_mp) as end_mp,
        a.spdlimit, b.surfwid
FROM
        (SELECT * from  speed_attrib  WHERE  spdlimit = 55)  a,
        (SELECT * from surf_width_attrib  WHERE  surfwid = 20)  b
WHERE  a.lrs_id = b.lrs_id AND
        NOT (a.begin_mp .>= b.end_mp  OR  b.begin_mp >= a.end_mp)
```

Next, we executed the following three-attribute query.

```
SELECT  a.lrs_id,
        GREATEST(a.begin_mp, b.begin_mp) as begin_mp,
        LEAST(a.end_mp, b.end_mp) as end_mp,
        a.lanes,  b.spdlimit,  b.surfwid
FROM
        (SELECT * FROM lanes_places, places  WHERE
                lanes_places.place_id = places.place-id  AND  lanes = ‘2’)  a,
        (SELECT  a.lrs_id,
                GREATEST(a.begin_mp, b.begin_mp) as begin_mp,
                LEAST(a.end_mp, b.end_mp) as end_mp,
                a.spdlimit,  b.surfwid
        FROM
                (SELECT *  FROM  speed_places, places  WHERE
                        speed_places.place_id = places.place-id
                        AND  speedlimit = 55)  a,
                (SELECT *  FROM  surface_width_places, places  WHERE
                        surface_width_places.place_id = places.place-id
                        AND  surface_width = 20) b,
        WHERE  a.lrs_id = b.lrs_id  AND
                NOT (a.begin_mp .>= b.end_mp  OR  b.begin_mp >= a.end_mp))  b
WHERE  a.lrs_id = b.lrs_id  AND
        NOT (a.begin_mp .>= b.end_mp  OR  b.begin_mp >= a.end_mp))
```

The speed of operation of this query was also virtually instantaneous. It was felt that on this basis we could be confident that the majority of DOT information queries could be resolved with no problems, given that nearly all would involve fewer than four attributes, with most involving only two.

2.4.4.2 Centralized Topology and Geometry Tests

At this point we had validated the concept of a “binary” database and the viability of the LRS Join. The next test was conducted to repeat essentially the same experiments using the place concept of centralized topology and geometry. To do so we modified the original SPEED_ATTRIBUTE tables, replacing the distributed topology and geometry with **Place** and creating a new PLACES table as follows.

SPEED_PLACES (**Place ID**, speedlimit)
SURFACE_WIDTH_PLACES (**Place ID**, surface_width)
LANES_PLACES (**Place ID**, lanes)
PLACES (**Place ID**, LRS ID, MP1, MP2)

As before, the SPEED_PLACES and SURFACE_WIDTH_PLACES tables had 73,635 and 89,041 records in each, respectively. The PLACES table had on the order of 174,884 records and the LANES_PLACES table contained 63,376 records. The same three-attribute query that was previously used to test the distributed topology and geometry was reissued on the centralized representation of the data. The form of the query is as follows.

```
SELECT  a.lrs_id,  GREATEST(a.begin_mp, b.begin_mp) as begin_mp,
        LEAST(a.end_mp, b.end_mp) as end_mp,  a.lanes,  b.spdlimit,  b.surfwid
FROM
        (SELECT * FROM lanes_attrib  WHERE  lanes = ‘2’)  a,
        (SELECT  a.lrs_id,  GREATEST(a.begin_mp, b.begin_mp) as begin_mp,
                LEAST(a.end_mp, b.end_mp) as end_mp,  a.spdlimit,  b.surfwid
        FROM
                (SELECT *  FROM  speed_attrib  WHERE  speedlimit = 55)  a,
                (SELECT *  FROM  surface_width_attrib  WHERE surface_width = 20) b
        WHERE  a.lrs_id = b.lrs_id  AND
                NOT  (a.begin_mp .>= b.end_mp  OR  b.begin_mp >= a.end_mp))  b
WHERE  a.lrs_id = b.lrs_id  AND
        NOT (a.begin_mp .>= b.end_mp  OR  b.begin_mp >= a.end_mp))
```

Again, despite the large number of records involved in the four tables that are accessed to satisfy the query, the results were the same – near instantaneous response. Essentially this query involves access to, and selection from, four different tables. Thus, it clearly demonstrates the speed with which ORACLE operates (even more so than the decentralized approach), effectively removing any doubts about database speed of operation – even on spatial queries!

3.0 KEY ISSUES

Three key issues were identified, discussed, considered, and resolved. These include: the centralization of the data and the physical networking, location, and distribution of data; temporal aspects of data; and, changes in the network or network topology.

3.1 CENTRALIZATION

ORACLE is to reside on specially purchased **database servers** located within the ETS Branch. These database servers physically store the databases and the ORACLE software to operate on the databases. The database server is to be connected to one or more **application servers** via high speed and high volume network connections. These application servers can be housed in various branches or units and it is upon these that stakeholder applications will run, however, the location of each application server will be dependent on the technical feasibility of maintaining the high-speed connection to the data server. This configuration maximizes performance for a well-designed application, since most of the network traffic will be between the application server and the database server. It is expected at this time that the applications will be written in Java with embedded ORACLE SQL commands in them.

Connected to each application server will be multiple client computers which simply run the interfaces to the application software. These interfaces are also expected to be written in Java, although some could be written in Visual Basic or C++. Typically, interactions between the clients and the application servers will be characterized by low bandwidth consumption, as the clients must wait on human input, whereas the interaction between application and database servers is automatic. Thus, the smaller client machines can efficiently process the receipt and display of data from the server as well as the gathering and conveyance of input data back to the server. The result is that stakeholder servers can be dedicated to running applications while the database server is dedicated to database processing. This will optimize the physical configuration given the availability of high speed, high volume connections between the applications and database servers.

In addition, the application server design permits “thin client” browser-based application interfaces published through world-wide-web (WWW) servers. Although these services will be used only by the stakeholders, publishing such an interface to the application permits access by users who may not have client programs installed on their computers such as county or municipal stakeholders, or use by users with scarce local resources such as laptop users and users in small field offices. These web interfaces use the same application server and database server as the client application program, and so can be produced for relatively little cost compared to web-enabling traditional two-tiered applications.

It is also expected that GIS software will run on the application servers and have direct access to the database server. This then provides stakeholders with the ability to access their data in either a tabular or graphical manner. GIS software in itself is highly developed and mature. However, its seamless interface with the ORACLE database is less so. As this maturity evolves it is expected that a GIS product will become available that will meet DOT needs and provide the seamless graphical and tabular data access. Users may also wish to use Personal ORACLE locally and retain full interoperability with the overall architecture.

3.2 TEMPORAL DATA FOR ARCHIVING

The size of the ultimate planning database cannot yet be estimated. Despite the fact that the database could grow to be very large there continue to be advances in digital storage capacity and media which can prevent size from becoming a problem. Problems or limitations on storage cannot at this time be seen for the statewide planning databases.

Given that all Statewide Planning databases will be stored in a centralized ORACLE database, one must consider the frequency and method with which the data will be archived. Three options are possible: snapshots, transactions, and a combination of these.

In the snapshot method the centralized database is always the “current” database. That is, the database represents the present. Periodically, a snapshot or copy is made of the database and stored or archived. Recent snapshots can be stored on line and more distant snapshots can be stored offline. The decision of how far back to go in time depends on the time dependence of the applications that use the data.

In the transactions method a record is kept of all the transactions performed on the database. That is, a record is kept of the changes made to the database. Transactions are reprocessed to rebuild the database as it existed at some previous time.

The recommended approach is a combination of the two. Snapshots should be taken at a periodic interval and stored along with all the transactions that occurred since the previous interval. This way a precise picture of the database is maintained for fast access, yet a continuous record of database activities is available for use if needed. The question of the time interval and frequency of archived snapshots cannot be determined at this time. To do so will require a clearer picture of the size of the database, an understanding of the demand level for archived data over time, and the optimality with which ORACLE can deliver the data.

3.3 TEMPORAL DATA FOR ATTRIBUTE VALUES

Time comes into play in two critical ways with respect to the attribute values stored in the database. First, *when* a data item received its value may be of interest. For example, when a road was renamed, when it was resurfaced, when a new highway was built, etc.

To deal with these questions from a database point of view requires consideration of the importance or criticality of time to a data item. Each data item must be examined to determine if time is a critical companion data item to it. If so, a date, time, beginning, or ending field(s) can be added to the table to associate a time value with the data item. This is one solution that can readily be adopted in the relational model when precise time is critical.

If the precision of time is less critical for a data item, enabling it to be measured in larger time quantities (like the nearest number of years), it may be possible not to explicitly store the time value. Rather, archived versions of the database could sequentially be searched until it is determined that the value of the attribute in question has changed. For example, if it is found that a roadway width has been 30’ wide in every 3 month archived version of the database back to and including January of 1996, then it is known that this width value is approximately 3 years and 6 months old (in June of 1999). Furthermore, it is also known that the roadway was recorded as having been widened in the fourth quarter of 1995.

Algorithms can be written to perform the sequential search and present the “age” to the user for this second solution. In the example given above the age that is determined is based only on the stored value at the time the snapshot was taken. If one wishes to determine the exact date the change took place that too can be done. It is done by restoring the database to an earlier snapshot point in time (say January 1996) and playing forward the transactions that occurred on the database until the value was changed. The exact date is thus obtained. In either of these scenarios the database need not be cluttered with date fields and values if it is not truly necessary. For most data this will be the optimal approach. But, as noted above, if time is critical and must be precise, a data field should be added to the table.

One advantage of maintaining such transaction information is that changes can be entered in advance of their effective date. This is common in financial applications. Often information in a transaction is received and committed to prior to the effective time of the transaction, and all parties wait to apply the changes to their master database until the effective time. This permits a transaction to proceed in spite of occasional outages on the part of one or the other participant, and also allows smoothing of the data entry demand curve.

To apply this concept to transportation, well-understood and anticipated events, such as the transfer of posted routes from one set of segments to another, could be entered ahead of time with a scheduled effective date, rather than behind-time as they're now handled. This would prevent us from having "rush jobs" (at least in certain areas) which interfere with the processing of more event-driven information. It's important to note that some customers of this data, including the Permits Section, will require data to be as near to real-time as is practical, since they will be making critical decisions on short time scales. This is a change from the traditional use of this data, wherein it has been published annually.

It should be noted that the focus of the database is on descriptions of characteristics of the highways themselves. Most attributes are thus related. One additional attribute that is of the same type identifies when the roadway was built. This information deals with the roadway network but it is an attribute of the topology of the network. For such an item of information, it is recommended that a separate database table be used to store this data. It too would be binary, including location and date. In this manner all new construction could be recorded and readily defined and identified. That is, time can be associated with the network as well as with the attributes. This issue is further discussed in Section 3.5.

3.4 CHANGES IN THE ROADWAY NETWORK GEOMETRY

The most critical element in the universe database is the combination of roadway network topology and geometry. To the extent possible, the initial implementation of the database should be as correct as possible with respect to the topology and mileages (geometry). However, both corrections and changes will need to be made to the database geometry as time passes. We do not refer here to the removal or addition of roads (changes to the linework in the context of adding or removing lines that represent the construction or retirement of physical roads), but rather to changes in geometric measures or references.

To properly document geometric changes and to accurately maintain the database geometry requires consideration of the measurement values associated with the geometric measures. As such, the recommended approach for NCDOT is to explicitly store values for distance references for each **Place** in the database. That is, given a roadway section with boundaries defined by **MP1** and **MP2**, we will explicitly store: **MP1**, **Measured From**, **MP2**, **Measured From**. The "measured from" fields will identify the physical network geometric locations from which an actual measurement was taken to determine the geometric locational values of **MP1** and **MP2**. Then, if changes occur to that geometric value, all the *places* related to that change can appropriately be adjusted.

Table 22 illustrates this point. It represents the LINEAR TOPOLOGY TABLE of Table 4 as well as the PLACES tables previously seen as Tables 10 and 15. (The former was the generic representation and the latter two were used to illustrate the example data from the appendix.)

Table 22 – PLACES

LRS ID	MP1	Measured From	MP2	Measured From	Place ID

In the previous tables we used **MP1** and **MP2** to delineate the boundaries of the roadway segment. Here we use **MP1** and **MP2** to do this so as to remove confusion. **Measured From** then tells us the origin for the values of **MP1** and **MP2**. That is, the values of **MP1** and **MP2** were measured from the **Measured From** location. Thus, not only do we know the values of **MP1** and **MP2** but we also know the origin, landmark, or anchor point (intersection, e.g.) from which they were measured.

Why is this useful? It is useful because field measurements will always be taken from a landmark, whereas LRS sections of highway are mile posted from an origin that can be a great distance away. Furthermore, if at some point in the future the landmark location itself is found to be inaccurate and must be changed, the "measured from" field identifies all other locations that must be changed by the same amount. The geometric integrity of the network is thus maintained.

This is a highly significant observation. Errors in field measurement are very possible, as are errors in data entry and maintenance. This **Measured From** field gives us the ability to correctly and easily find and adjust all measurements that are incorrect as a result of depending on a reference point that is itself incorrectly located.

How would this work? Assume node C2 was incorrectly measured to be at **MP** 68.55 and that instead, its correct measurement is found to be at 68.35. This is a change in the geometry of the topology of the network; not just in the location of an attribute. This correction can easily be made in the INTERSECTIONS table for node C2. But, in addition, anything that has ever been measured from C2 must also be corrected. How? Simply by searching every **Measured From** field in the database to find all other values that were actually measured from that node. The PLACES table shows us that **Place ID** 17 was measured from C2. If C2 was found to be 0.20 miles in error, then so too is this **Place ID** 17 **MP1** also 0.20 miles in error. It can easily be changed from **MP1** 69.20 to **MP1** 69.00. Thus, corrections to the entire database are easily accomplished.

3.5 TIME AND THE NETWORK

There are three aspects of the concept of "network" to consider. First, the network is represented using lines – either on paper, maps, or in CAD, for example. Second, the lines represent physical roads. Third, portions of the road are identified as places.

We do not wish to associate time with lines. That is, when a line was added, removed, or changed is of little consequence and should not explicitly be treated in the database. However, it may be of interest to associate a measure of time with a physical roadway segment. Two approaches to do so emerge: associate time with places, or create a separate new table that handles time in its appropriate context.

The two approaches would result in the following two database design alternatives. First, an **Effective Date** is added to our POSTED ROUTES and PLACES tables as follows.

PLACES (LRS ID, MP1, Measured From, MP2, Measured From, Place ID, Effective Date)
POSTED ROUTES (LRS ID, MP1, MP2, Posted Route, Effective Date)

But this approach essentially associates time with lines rather than with roads and is thus not recommended. Rather, what is recommended is to create a table similar to the following.

CONSTRUCTION DATE (Place ID, Effective Date)

In doing so we explicitly identify newly built roads that are added to the database as they are built, or, for existing roads we associate a construction date with them. Physically, this also means that we have fewer records than if we chose the previously mentioned alternative. In the first approach we would be adding fictitious dates to places and routes that did not represent when the road was built, but represented instead when the lines were identified in the database. This is unnecessary and the second approach is recommended.

4.0 EXISTING UNIVERSE DATA

4.1 BACKGROUND

The ***GIS Unit*** is responsible for the development and maintenance of GIS graphic files, as well as several mainframe files. The GIS ***Road Inventory Section*** is responsible for managing the ***Universe File***, a segmented database for the approximately 78,000 miles of state-maintained roads, and the ***Feature File***, which contains road intersection and landmark features. These two files were developed to completely describe the road network. For example, a segmented link in the Universe File may contain several intersections that must be recognized as features, e.g. for turning movement studies or accident reports.

As mentioned previously, the Universe File is a data file that contains information related to roadway characteristics. The Universe File contains approximately 4900 sections that are maintained as the Federal Highway Administration's (FHWA)/Highway Performance Monitoring System's (HPMS) sample data.

In addition to the Universe File is the ***Supplementary File*** that has similar attributes and can be considered an extension of the Universe File for approximately 4900 HPMS sample sections. These characteristics are currently referenced according to segments described in milepost and section length measurements. This file has been combined with GIS graphics to allow elemental display and analysis of different types of transportation data. The Feature File is a complimentary data file, which contains information related to point features located along the routes stored in the Universe File. The data stored in both these files is made available by the GIS Unit to other units for their own analysis purposes.

4.1.1 Highway Performance Monitoring System (HPMS)

The Highway Performance Monitoring System (HPMS) is a continuous data collection effort that was developed and implemented in a joint effort between the States and the FHWA. Today, the HPMS has grown to contain over 110,000 sample section segments nationally and is the most complete data system used to evaluate the physical condition and usage of the country's transportation infrastructure.

There are over 4900 sample sections in the North Carolina portion of the system. The Federal government uses the HPMS database as a primary source of information about the Nation's highways. It serves as both a national and statewide information clearinghouse that incorporates all of the nations public roads. It also acts as an analytical simulation system that can be used in a variety of situations. The HPMS also contains detailed sample data for rural, small urban, and urbanized areas within each state at established levels of precision.

The State Highway Agencies, local governments, and metropolitan planning organizations all work together to collect and submit the required information contained in the HPMS system. The FHWA is responsible for identifying the data to be collected, establishing the most efficient data collection methods, developing proper analytical procedures, and analyzing the data. All of these activities help to facilitate better highway planning, policymaking, and decision making at the Federal level. The HPMS has always been an important part of policy planning, but in the last 20 years, the FHWA has relied more on the HPMS in making program administration decisions.

The HPMS database is unique in that it directly relates many roadway characteristics, including physical geometry, operational data, usage (travel), pavement conditions, and performance data. This data can be analyzed and summarized at the local, statewide, regional, and national level with respect to a specific highway system. A new GIS capability should greatly enhance the user's ability to both display and analyze HPMS data. HPMS analytical models have been modified so that individual State transportation agencies can take advantage of the data to help evaluate the condition and performance of their own highway system.

4.2 MAJOR APPLICATIONS

The major applications performed by the ***GIS*** unit include the following:

1) Maintenance of Universe, Supplementary, and Feature Files.
2) Production of the annual HPMS report.
3) Generation of Maps on Microstation.

These applications depend on the data that presently resides in the existing data files as enumerated below.

4.3 DATA – CONCEPTUAL DESCRIPTION

In this section, data from the Universe File, Supplementary File, and the Feature File are listed and described in a conceptual manner. That is, all data items are not explicitly enumerated and described, rather the data types are characterized. Data required for each application may be classified into several types of categories, including:

- Identification
- Classification
- Spatial
- Design Criteria
- Traffic Statistics

The following is a conceptual presentation of the data currently produced by the ***GIS Unit***. The applications are listed along with a general description of the data types included in the data files:

- ***Universe File***: identification, location, geometric, classification, demographics, historical, traffic statistics.
- ***Supplementary File***: identification, location, geometric, classification, traffic statistics.
- ***Feature File***: identification, location, geometric, classification.

4.4 EXISTING DATA ITEMS

In this section, specific data items (attributes) are enumerated for each data file as they currently exist. These data items originate from various sources within NCDOT. The data files are intended to provide a comprehensive summary of information related to the roads and the point features found along the roads. Definitions for each data item can be found in ***NCDOT Code Manuals I, II, and IV***. Refer to these documents for a complete explanation of the data enumerated below. The purpose of including this section in the report is to document the data as it was prior to the development of the database. Section 4.5 presents the data as it is to be incorporated into the new universe database.

The Universe File contains the following data items in a data file format. All of the data items are currently stored in a coded alphanumeric format. See ***Code Manual I*** for a detailed description of the data item and coding format.

- I.D. Number
- County
- Route
- List Control
- Mile Post
- Section Length
- Inventory Control
- Terminal Description
- State System
- National Highway System
- Functional System
- Federal Domain
- Division
- Town
- Population (Municipal)
- Terrain
- Sight Distance %
- Weighted Design Speed
- Urban Identification
- Population (Urban Area)
- Record Continuation Code
- Right-of-Way Width
- Access Control
- Speed Limit
- Year of Improvement
- Type of Improvement
- Number of Travel Lanes
- Surface Width
- Surface Type
- Left Shoulder Width
- Left Shoulder Type
- Right Shoulder Width
- Right Shoulder Type
- Median Width
- Median Type
- Year Added to State System
- Truck Percentage Code
- Design Hour Speed
- Truck Route Designation
- Average Daily Traffic
- P.T.C. Count Station
- Year of Traffic
- Sample Link Number
- Interstate Milepost
- Percent of Trucks Off-Peak
- Traffic Growth Factor
- Parking Left Side
- Parking Right Side
- Turning Lane(s) Width
- Urban Location
- HOV Lanes
- Surveillance Systems

The Supplementary File contains the following data items in a data file format. All of the data items are currently stored in a coded alphanumeric format. See ***Code Manual II*** for a detailed description of the data items and coding format.

- Identification Number
- County Number
- Numbered System
- Route Number
- Milepost
- Sample Link Number
- Sample Link Number Subdivision
- Section Length
- Pavement Section
- Structural Number
- Peak Capacity
- Directional Factor
- Type of Signalization
- Typical Peak Percent Green Time
- Drainage Adequacy
- Type of Development
- Number of Grade Separated Interchanges
- Number of at-Grade Intersections with Public Roads
- Signals
- Stop Signs
- Other or No Controls
- Left Shoulder Width
- Widening Feasibility
- Horizontal Alignment Adequacy
- Vertical Alignment Adequacy
- Concrete Joint Spacing
- Load Transfer Devices
- Type of Base
- Overlay or Pavement Thickness
- Type of Subgrade
- Subsurface Drainage
- Turning Lanes
- Interstate Milepost
- Curves by Class
- Grades by Class
- Future AADT
- Future AADT Year
- Percent Peak Single Unit Trucks
- Percent Average Daily Single Unit Trucks
- Percent Peak Combination Trucks
- Percent Average Daily Combination Trucks
- Number of at-Grade Railroad Crossings
- Curves by Class
- Grades By Class

The Feature File contains the following data items in a data file format. All of the data items are currently stored in a coded alphanumeric format. The data items are contained in several different record types, which are also enumerated below. See ***Code Manual IV*** for a detailed description of the data items and coding format.

County Record:

- County Number
- Identification Number
- Record Type
- Correction Type
- Begin County
- End County
- County Name Abbreviation

Municipal Record:

- County Number
- Identification Number
- Record Type
- Correction Type
- Begin Municipality
- End Municipality
- Municipal Code
- Municipal Name

Route Type:

- County Number
- Identification Number
- Record Type
- Correction Type
- Begin Route
- End Route
- Route Type
- Special Routes
- Couplet Direction Code
- Route Number (Primary, Secondary, Municipal)
- Milepost

Coinciding Section Record:

- County Number
- Identification Number
- Record Type
- Correction Type
- Coinciding Routes

Item Record:

- County Number
- Identification Number
- Record Type
- Correction Type
- Item Identification
- Special Routes
- Couplet Direction
- Item Type (Interstate Mile Marker, Primary Route, Secondary Route, Municipal Street, County Line, Municipal Line, State Line)
- Distance to Next Item
- Direction to Next Item
- Intersection Type
- Loop Condition
- Area Descriptor

Special Item Record – Structure:

- County Number
- Identification Number
- Record Type
- Correction Type
- Type of Structure
- Bridge Number
- Distance to Next Item
- Is Structure Over/Under Inventoried Route
- Route Number or Name of Item Crossed

Special Item Record – Railroad Grade Crossing:

- County Number
- Identification Number
- Record Type
- Correction Type
- Type of Structure
- Railroad Crossing ID Number
- Distance to Next Item
- Railroad Company Code

4.5 PROPOSED DATA ITEMS

In preparation for the database design portion of the project the existing universe data items have been reviewed to consider their relevance in the new universe database. Some redundant data items have been identified and removed. Some data items have been reclassified considering their proper ownership within a unit. The following enumeration of universe data items reflects a more focused and directed categorization of the items. This collection of data items from the Road Inventory Section is to form the core universe database when the schema is designed.

Data Generated/Owned by Statewide Planning – GIS Road Inventory Section

- LRS Route Number and LRS Milepost
- Secondary Route Number and Milepost
- List Control (couplet indicator)
- Inventory Control (directional splits on divided highways)
- Terminal Description
- State System Coding (rural secondary, municipal secondary, etc.)
- Federal Domain (special federal designations)
- Highway Division Code
- Terrain
- Record Continuation Code
- Year of Improvement to Section
- Type of Improvement to Section
- Number of Travel Lanes
- Surface Width
- Surface Type
- Left Shoulder Width
- Left Shoulder Type
- Right Shoulder Width
- Right Shoulder Type
- Median Width
- Median Type
- Year Added to State System
- Sample Link and Subdivision Number
- Parking (Left and Right Sides)
- Turning Lane(s) Width
- High Occupancy Vehicle Lanes
- Surveillance Systems
- Drainage Adequacy
- Number of Grade Separated Interchanges
- Number of At-Grade Intersections
- signals
- stop signs
- others
- Median Shoulder Width
- Widening Feasibility
- Horizontal Alignment Adequacy
- Vertical Alignment Adequacy
- Overlay or Pavement Thickness
- Turning Lane(s) Configuration
- Curves by Class
- Grades by Class
- Passing Sight Distance Percentage
- Right-of-Way
- Type of Access Control
- Concrete slab thickness
- Concrete joint spacing
- Load Transfer Devices
- Type of Subgrade
- Subsurface Drainage Type

Data Generated/Owned by Statewide Planning – GIS Mapping Section

- Town Code
- Town Population Code

Data Generated/Owned by Statewide Planning - Planning Sections

- National Highway System Designation
- Functional Classification System
- Urban Identification Code
- Urban Population Code
- Urban Location
- Peak Capacity of Intersection
- Type of Development

Data Generated/Owned by Statewide Planning – Traffic Forecasting

- Design Hour
- Directional Factor of Traffic
- Future AADT
- Future AADT Year
- Traffic Growth Factor

Data Generated/Owned by Statewide Planning – Traffic Surveys

- Truck Percentage
  - Off Peak Hour (Single Unit and Combination Trucks)
  - Peak Hour(Single Unit and Combination Trucks)
- Annual Average Daily Traffic (AADT)
- AADT Year
- Portable Traffic Count Station Id#

Data Generated/Owned by Traffic Engineering Research

- Primary Rout Number
- Truck Route Designation
- Type of Signalization
- Peak % Green Time
- Number of at-grade Railroad Crossings
- Speed Limit

Data Generated/Owned by Pavement Management Branch

- Structural Number of Asphaltic Concrete Pavement
- Pavement Roughness
- Pavement Condition

5.0 RECOMMENDED UNIVERSE DATABASE SCHEMA

The design for the universe database schema is to be, for the most part, binary. That is, a large number of universe tables will have exactly two attributes. In general these will have the form:

TABLENAME (Place ID, Attribute Value)

This form was previously illustrated by Table 16 and will be further illustrated in the following section.

There are a number of reasons for selecting this approach. First and foremost is the ease of maintaining each attribute independent of each other. The process of finding information related to a single attribute and changing it will significantly improve when only one database table needs to be accessed for that attribute value.

Second, it is extremely easy to join one or more of the binary tables to satisfy queries or data analysis requests. The LRS Join was demonstrated to make this a nearly real-time process and no difficulties are expected in implementing the join, given the small size of the tables.

Third, it is easier to link to and transport selected data from binary tables to the GIS. Individual attributes can easily be made to correlate directly with a GIS coverage. The GIS coding for analysis application development can also be very much simplified, thus increasing programmer efficiency and Unit output while providing a degree of integrity assurance regarding the data.

5.1 SPATIAL TOPOLOGY

The actual spatial topology implementation differs somewhat from the conceptual outline discussed earlier in the report. To best illustrate the base spatial topology and geometry the example roadway in the appendix is fully presented in this section. The tables to represent the spatial and geometric data of this roadway network follow.

Table 23 – PLACES

LRS ID	MP 1	Measured From	MP 2	Measured From	Place ID
515	0.00	0.00	1.00	1.00	1
623	68.00	68.00	71.50	71.50	2
624	105.00	105.00	108.50	108.50	3
742	25.00	25.00	26.80	26.80	4
742	25.00	25.00	27.25	27.25	5
623	68.00	68.00	69.40	69.90	6
623	69.40	69.90	71.50	71.50	7
624	105.00	105.00	106.70	106.90	8
624	106.70	106.90	108.50	108.50	9
742	25.00	25.00	26.45	26.45	10
742	26.45	26.45	26.80	26.80	11
742	26.80	26.80	27.25	27.25	12

515	0.00	0.00	0.50	0.00	13
515	0.50	0.00	1.00	1.00	14
623	68.00	68.00	68.55	68.55	15
623	68.55	68.55	69.20	68.55	16
623	69.20	68.55	70.80	71.50	17
623	70.80	71.50	71.50	71.50	18
624	105.00	105.00	105.45	105.45	19
624	105.45	105.45	106.20	106.90	20
624	106.20	106.90	108.50	108.50	21
742	25.00	25.00	25.70	25.70	22
742	25.70	25.70	27.00	26.80	23
742	27.00	26.80	27.25	27.25	24
624	105.00	105.00	108.50	108.50	25
623	68.00	68.00	71.50	71.50	26
742	25.00	25.00	27.25	27.25	27
624	105.45	105.45	108.50	108.50	28
623	68.55	68.55	71.50	71.50	29

Table 24 - INTERSECTIONS

LRS ID	LRS MP	Node ID	PR ID	PR MP	County
624	105.00	N1	54	36.8	Y
624	106.90	N4	54	1.45	X
624	108.50	N8	54	3.05	X
623	68.00	N2	78	42.90	Y
623	69.90	N5	78	1.35	X
623	71.50	N9	78	2.95	X
515	0.00	N6	115	14.30	X
515	1.00	N10	115	15.30	X
742	25.00	N3	65	12.50	X
742	25.00	N3	115	12.50	X
742	25.70	N4	65	13.20	X
742	25.70	N4	115	13.20	X
742	26.45	N5	65	13.95	X
742	26.45	N5	115	13.95	X
742	26.80	N6	65	14.30	X
742	26.80	N6	115	14.30	X
742	27.25	N7	6	14.75	X

COUNTY BOUNDARIES

LRS ID	LRS MP	Node ID	PR ID	PR MP	County
624	105.45	C1	54	0.00	X
624	105.45	C1	54	37.25	Y
623	68.55	C2	78	0.00	X
623	68.55	C2	78	43.45	Y

Table 25 – POSTED ROUTES

Place ID	Posted Route ID	Posted MP1	County 1	Posted MP2	County 2
1	NC 115	14.30	X	15.30	X
2	NC 78	42.90	Y	2.95	X
3	NC 54	36.80	Y	3.05	X
4	NC 115	12.50	X	14.30	X
5	NC 65	12.50	X	14.75	X

COUNTIES

Place ID	County
1	X
5	X
15	Y
19	Y
28	X
29	X

Table 26 - RAILROAD CROSSINGS

LRS ID	Milepost	Measured From	RR ID
623	69.20	68.55	1
624	106.20	106.90	2

Note that the INTERSECTIONS table contains only intersections between roadway network *lines*. Even though county boundaries are also nodes they are separately stored in another table in a manner similar to other point data like railroad crossings. This enables us to "double store" them so that their dual PR MP values can be given. The COUNTIES table enables us to locate every place that is fully contained within a given county. No places which cross county boundaries can be stored in this table. Likewise, the INTERSECTIONS table enables us to locate every node that is contained within a given county. There is one expected addition to the INTERSECTIONS table. It must be known if the RR MP numbering was in the same or the opposite direction of the LRS MP numbering. Thus, an added column is expected to account for this need.

A number of observations are in order. First, the general topology of the roadway network is captured using these tables and the concepts embodied in the NCDOT base LRS. The tables are compact and implement efficiently. As discussed earlier, they centralize the topology and geometry to enhance maintenance. They also contain appropriate references to adjust the geometry if corrections need to be made and they allow this to be done without jeopardizing the integrity of the database. Finally, they contain the appropriate correlation to the countly/route/milepost referencing scheme. This is a direct representation that allows appropriate transformations between the schemes.

The data contained in these tables is shown for illustrative purposes. The data is derived from the example network shown in the appendix.

5.2 UNIVERSE ATTRIBUTE DATA

The universe data, as mentioned earlier, is to be represented mostly in binary tables. This is one of the most important decisions made by the committee regarding the database. It has been justified on the basis of the speed of operation of the Oracle database and the ability of the users to maintain database integrity through separation of attribute values in different tables.

Again, consider the roadway network from the Appendix as an example. Illustrative tables to represent the attribute data of this roadway network follow. Only a few tables are shown here. In fact, we use the ones that have previously been used throughout the text. The final tables to be implemented in the database should be designed in the next phase of the project. Section 5.3 discussed the database design task further.

Table 27 – PAVEMENT CONDITION

PlaceID	Pavement Condition
1	Good
6	Moderate
7	Good
8	Good
9	Excellent
10	Poor
11	Moderate
12	Poor

Table 28 – PAVEMENT WIDTH

Place ID	Pavement Width
13	26’
14	28’
15	26’
16	28’
17	26’
18	28’
19	26’
20	28’
21	26’
22	24’
23	26’
24	28’

Table 29 – CONSTRUCTION DATE

PlaceID	Date
25	6-4-21
26	8-1-36
27	3-17-17

Note that only three attributes are represented here, but the methodology for their representation is clear. It is, in fact, representative of how the universe database tables will be structured.

It should no noted, however, that not all universe attribute tables will be binary. Careful consideration will be given to each data item during the database design process. It is expected that a number of data items will be found to be logically related and thus should be brought together in a single table. At the same time careful consideration will be given to data maintenance as well so that the arity of each table supports both logical design and ease of maintenance.

5.3 DATABASE DESIGN TASK

Design, development deployment, enhancement, training, and extension identify the highest level database task activities to be undertaken. The following outline summarizes, in abbreviated form, the activities to be undertaken in these areas. This enumeration is included herein as a guide only. A more detailed activity list is necessary before beginning work on any individual task. Yet this enumeration can serve as a reference for understanding the scope of the work remaining. Even then, it is to be understood that continued enhancement, refinement, and modification of the system will occur.

A. Design
    1. Database schema design

        1.1 Global design
            a. solve topological problems unique to a stakeholder

        1.2 Detailed design
            a. design geometry and attribute database structure for each stakeholder
            b. assess field procedures and their impact

        1.3 Data inter-relationships – integrity constraints
            a. Identification
            b. Documentation
                1. equations and other
                2. logical relationships
                3. procedures and algorithms

    2. Database management system design

        2.1 User Interface – Data Definition Language (DDL)
            a. DBA functionability
            b. Stakeholder functionality

        2.2 User Interface – (DML)
            a. Basic operators

LRS Find
LRS Join

- b. Topology Operators
  - 1. insert, delete, change
- c. Geometry Operators
  - 1. insert, delete, change
- d. Attribute Operators
  - 1. insert, delete, change
- e. Filters to convert to/from location referencing methods
  - - segment data (place)
  - - point data
  - 1. CRMP to LRS
  - 2. LRS to CRMP
  - 3. L.N. to LRS
  - 4. LRS to L.N.
  - 5. intersection, offset, direction to LRS
  - 5. LRS to IOD

2.3 User Interface – Data Query Language – (DQL)

- a. Design code for ad hoc queries for each stakeholder
  - 1. topology
  - 2. geometry
  - 3. attributes
- b. Report design for each stakeholder

2.4 User Interface – Application Program Interface (API)

- a. Design interface to operate, control and supply data to applications
- b. Identifying, documenting, other major inventory applications.
- c. Design application to generate the single universe table from which the HPMS reporting data is generated.

B. Development

1. Coding

1.1. Schema

- a. Schema – attribute data
  - - universe file converts to universe table
  - - Universe table converts to universe database
  - 1. primarily binary tables
  - 2. multi-attribute tables where appropriate
  - 3. tables compressed from L.N. representation to LRS
- b. Schema – spatial data
  - - convert universe key to LRSID and mileposts
  - - generate all spatial tables

        - 1.2. User interfaces manipulation capability to use the database
            - DDL
            - DML
            - DQL
            - API

        - 1.3. Applications
            - Filters
            - HPMS
    - 2. Testing - In house testing to verify software functionality and performance using full universe data set.

C Deployment to each stakeholder as appropriate and as development proceeds.

D Enhancement
- 1. Evaluation
    - -.performance
    - - function
    - - omissions
- 2. Redesign
- 3. Redevelopment
- 4. Redeployment

E Training
- 1. Basic database concepts
    - 1.1 Conceptual LRS
    - 1.2 DB and DBMS theory
    - 1.3 LRS Implementation in the DB
- 2. LRS Implementation in the GIS
- 3. Universe database & information management system use

F. Extension
- 1. Assist other Units in adopting the design tools that were developed

6.0 KEY RECOMMENDATIONS

It should be emphasized that this report presents a template database. The real database will not exist until it is actually built. Therefore, this report should be viewed as a roadmap for the construction of that database. It should also be viewed as a source of answers to critical questions that have arisen during the development of the LRS as well as a source of solutions to problems that have also been identified.

The **single most important recommendation** of this report is for NCDOT to formally embark on a database design effort, including final attribute data schema design, filter program design, standard data query definition, standard data maintenance operator definition, report generation design, data entry form design, etc.

A key consideration is the recognition that there is a distinct difference between the topology, geometry, and attribute data that must be stored in the database. Furthermore, it is critical to clearly articulate and design for the ability to insert, delete, and modify all of the attribute data, geometric data, and the network topology itself. Particularly careful attention must be paid to operations that change the topology or geometry of the network, as these have critical ramifications for all database users. Additional consideration must be given to the database operations to be carried out by stakeholders, by Roadway Inventory, and by Engineering Technology Systems. It is recommended that the database designer have significant database expertise and on-site presence.

The following is a compilation of recommendations that have arisen in the preparation of this report. Some of these issues will require much thought and deliberation before any action is taken. The recommendations are *not* presented in any order of importance or significance.

1. The Ordinance database and the Universe File should be integrated so that the similar data these sources contain is located in one place. This will eliminate duplicate data items and data discrepancies. The Ordinance database is more accurate than the Universe File in terms of recording actual field characteristics. Adopting the Ordinance database's level of accuracy will help to serve those requesting data in a much better fashion.

2. All units should seriously consider converting their relational databases to ORACLE as soon as possible in order to make the transition to the universe database easier and more universally accepted. There are several databases currently in use, including ACCESS, ORACLE, PARADOX, DBASE, etc. Switching to ORACLE is not a monumental task since each database program operates on the same basic principles and uses ***Standard Query Language (SQL)*** as a base. Converting to ORACLE now will enable personnel to become familiar with the program and facilitate the future transition to the universe database.

3. All stakeholders participating in the universe database should, over time, adopt the standard unique ID and milepost topological structure and convert

their data to fit both the relational model and the proposed LRS structure. This report has provided a basic relational structure that can be used as a template or as an internal structure for their data. These relational structures can be followed as a guideline and modified to support each unit's needs.

4. There should be an overall examination of existing databases across business unit boundaries to identify and eliminate the duplication of data. Data duplication may occur in several areas, including data collection, data processing, and data storage. Identifying duplicate data items will increase the overall efficiency of the data correction process. That is, when changes must be made to the system, these changes will only have to be made in one place.

5. Temporal data needs for *archiving* data should be met using a combination of database snapshots and transactions. After a period of time, wherein the system is completely built and experience is gained using it, this approach can be reconsidered.

6. Temporal data needs for *attribute values* should be handled on a case by case basis when the database is designed. If time is critical to an attribute it should be explicitly stored as a companion attribute. Otherwise, it should not be stored.

7. The database is to be physically centralized and maintained on the ETS database server.

8. The topology and geometry of the roadway network should be maintained by the Roadway Inventory Section of the GIS Unit. All attribute data should be maintained by each of the stakeholders who currently maintain that data, unless they do not have the internal manpower or resources to do so. In that case, the data should be maintained by either ETS or Roadway Inventory.

9. Measurements by field personnel should all be made from intersections or county boundaries. These are the "anchor points" of the linear referencing system and measurements thus taken can easily be converted to the LRS and to other measurement systems for inclusion into the database.

10. Training should be established for stakeholder users of the LRS and the database. Topics covered should include LRS concepts, the NCDOT LRS, basic relational database concepts, the NCDOT database, representations of topology and geometry in databases, etc.

11. The LRS implementation should utilize the centralized topology (PLACE) concept to allow for centralized storage and maintenance of the topology and geometry.

APPENDIX

The figures on the following pages illustrate an example roadway network for purposes of illustration. Included are four main routes, a county boundary, a railroad, and coincident routes.

Figure 1 illustrates the posted route numbering and the posted route mile posting. It includes numbering for the intersections to identify these topological nodes. The intersections include the intersection of the county boundary with the roadway network but these nodes are identified with a different beginning character (C instead of N).

Figure 2 shows the corresponding LRS routes and the LRS mile posting for the same network. The details remain the same otherwise.

Figure 3 utilizes the same network to now show the pavement condition of the roads. Thus, an attribute value is illustrated.

Figure 4 shows another attribute value – pavement width. All other notations are the same as in Figure 2.

Figure 5 shows the relationship between pavement condition and pavement width. The figure overlays Figures 3 and 4 to obtain a resulting figure that tells the viewer what the relationships are between these attributes.

Figure 6 is an additional figure that provides data about count stations. The figure identifies locations for 11 count stations and provides information regarding the use of those count stations.