Functional Specifications of the Pavement Integrated Data Base System for Pavement Management

DIMITRI A. GRIVAS, YUNG-CHING SHEN, R. PETER FROSCH, AND RICHARD GARRABRANT

The functional specifications of the Pavement Integrated Data Base System (PIDBS) of the New York State Thruway Authority are described. The PIDBS serves as the primary mechanism of linking all applications in the domain of pavement management. Its structure follows a top-down design that encompasses all functions within the pavement management process. The design of the data base uses the entity-relationship data modeling technique, in which data elements are viewed in the form of entities having defined keys and attributes and relationships among these entities. Each data element is defined and documented in a data dictionary. The data base has been implemented using the ORACLE relational data base management system. Applications include interactive interfaces with dedicated screens, forms, and menues for report generation and programmatic interfaces with automatic data conversion for analysis purposes. A prototype version of the PIDBS has been tested and further expanded to serve as a working model of the system. Important aspects of the development effort, tools used, and system applications are discussed. Establishment of functional specifications, which clearly define the structure and capabilities of the system, is an important requirement for a successful data base design.

The primary motivation for the development of data base systems is the need to support pavement management activities through effective storage, retrieval, and analysis of data. Satisfying this need should be an important objective of the overall effort that aims to develop and implement a comprehensive pavement management system (PMS) (1).

The usefulness of data base systems is derived from their ability to provide an efficient environment for data processing. They can serve a variety of functions that range from central data storage (for data sharing among users) to automated report generation and analysis through program interfaces. They can also satisfy the information needs at all levels of management. In this sense, data base systems may be thought of as integral components of decision support systems, which are broader in scope.

A recent study by Haas and Hudson (2) has identified several promising technologies for future advances in pavement management, on the basis of 20 years of experience in developing PMs. Although each of these developing technologies aims to assist in a distinct manner the decision process of finding cost-effective strategies in pavement preservation, the underlying theme and important requirement for their success is the presence of a data base system. Other studies (3,4) have also pointed out the need for information and techniques used in pavement management to be coordinated and integrated with the aid of data base systems.

Data base tools have evolved dramatically over the past few years. Their original capabilities, limited to passive repositories of data, have been greatly enhanced by the availability of modern relational data base management systems. These have made it possible to develop integrated data base systems (5,6) that incorporate all features of the decision process and utility programs within the boundaries of pavement management.

The broad goal of the present research effort is to develop a pavement integrated data base system (PIDBS) that supports the pavement management activities at the New York State Thruway Authority (NYSTA). The system aims to satisfy the needs of individual users within the pavement program of the NYSTA, and at the same time, enable intelligent external interaction with a variety of application programs.

Tasks associated with the functional design of PIDBS are described. These tasks include system analysis, data modeling, data dictionary, and data base applications. Also presented is a prototype version of the PIDBS, which has been further expanded to serve as a working model of the system.

OVERVIEW OF PIDBS

System analysis for the development of the PIDBS was initiated by examining the decision process, reading available documents, and interviewing personnel responsible for the pavement program of the NYSTA. The structure of the system was synthesized from three major pavement management activities, namely data collection, project level analysis, and network level analysis. A CASE tool was employed to describe the top-down design of the system, and the functional structure of the system components.

The entity-relationship (E–R) data modeling technique (7) has been extensively used for the purposes of data analysis. Main tasks were (a) identification of entities, keys, and attributes from data sources associated with each system component; and (b) establishment of entity relationships, and justification of the functionality of these relationships, on the basis of a semantic description of the pavement management

D. A. Grivas and Y.-C. Shen, Department of Civil Engineering; R. P. Frosch, Information Technology Services; Rensselaer Polytechnic Institute, Troy, N.Y. 12180. R. Garrabrant, New York State Thruway Authority, Albany, N.Y. 12209.
process. Results of data analysis included definitions of all data elements, detailed data documentation in a data dictionary, and data illustrations through E–R diagrams.

The data base has been implemented using the ORACLE Relational Data Base Management System (RDBMS) (8). Data base applications can be achieved through either interactive or programmatic interfaces. Reports, forms, and menus are examples of interactive applications that are generated through standard Structured Query Language (SQL) (9) commands within the ORACLE tool environment. Life cycle cost analysis, project-level analysis, and network-level analysis are examples of application programs that are developed with procedural languages, such as C, Pascal, and Fortran. The ORACLE programmatic interface allows the developed data base system to be integrated with application programs.

BACKGROUND

The Pavement Management Concept

Pavement management involves a coordinated use of available resources to achieve desired goals (2,10). It can be effective only if the necessary information and decision-making methodology are available. In this sense, the general requirements for pavement management are not different from those of other areas of management.

The present development effort aims to produce a system that encompasses a wide range of activities, which include the following (4):

- Data collection, handling, and processing;
- Establishment of criteria for decision making;
- Analysis and evaluation of project alternatives;
- Selection of optimal actions; and
- Implementation of the pavement program.

Decisions associated with each of these activities are made at project and network levels. A good management system requires a planned, coordinated execution of these activities as well as the inclusion of other elements, such as pavement condition and deterioration models, life cycle cost analysis, and others. The PIDBS is expected to serve needs for data entry, reporting, and queries associated with these activities. Furthermore, it is expected to act as a linking mechanism among the application tools used at various stages in the decision process.

Integrated Database Model

Among the first conceptual models of an integrated data base system was the one developed during the 1985 Woods Hole Workshop (5). This model aimed to support the entire range of applications involved in architectural planning and engineering design and construction of buildings. A prototype version of the system included the following components:

- Relational DBMS and query language,
- Two or more application systems,
- Computer hardware capable of supporting the selected DBMS and applications,
- Engineering workstations with graphics capabilities, and
- Integration software.

In a similar manner, the PIDBS consists of a central data base and application interfaces, integrated so as to allow data manipulation by either nonprocedural or procedural methods. The central data base is mainly operated by nonprocedural query methods that provide end users the capability to pursue the functions of data insertion, update, deletion, and retrieval. Application interfaces are designed as procedural languages, in which the programmer can incorporate the data base query statements and instructions needed to access and manipulate data.

RDBMS Tool

The basic structure of the employed ORACLE RDBMS (8) is a two-dimensional table. Data retrieval is achieved through the SQL data manipulation language. SQL, which was first developed by Chamberlin et al. (9) at the IBM Research Laboratory, has become the standard language of relational data base management systems. Although shortcomings of the language have been recognized (11,12), its popularity has not diminished. The use in the present work of an RDBMS based on SQL has the important advantage of portability, as there are many commercial systems that use SQL for data manipulation.

SYSTEM ANALYSIS

Functional Structure of PIDBS

Although pavement management has clearly defined objectives, the data requirements to achieve these objectives are not always easily defined. Structured analysis techniques can be helpful on this subject, as they enable identification of data elements and definition of data structures (e.g., in the form of data entities). Two specific techniques used in this study are (a) data flow diagrams, and (b) hierarchical modules for data applications.

The data base requirements for the pavement management system of the NYSTA have been presented by Grivas (4). The general data flow in the pavement management process is shown schematically in Figure 1. Each rounded rectangle appearing in the figure represents a specific process (e.g., computing, analysis, or data processing); each flat rectangle with an open end represents a data store (i.e., data files or data base); and each square box represents an external entity that identifies either data origin (outside the system) or data destination (after processing).

The functional structure of the PIDBS is shown in Figure 2. The data base provides a central pool of data for pavement design, pavement condition determination, maintenance records, traffic volume, climate records, etc. The figure also shows the data base links to project level analysis, network level analysis, damage assessment methodology, life cycle cost analysis, preferential rating analysis, annual program analysis, and traffic–safety control analysis.

As expected, different data base functions require different amounts of data. For example, the function of pavement con-
FIGURE 1 General data flow diagram of pavement management process.

FIGURE 2 Preliminary functional structure of PIDBS.
dition determination requires a large amount of data. This is not the case for the function of safety control. Also, not all data elements stored in the PIDBS are expected to be accessed with equal frequencies. Finally, some data elements may be changed over time (or even eliminated) as requirements for information change, whereas others may possess historic value and, thus, continue to be stored in the data base.

Data Collection

Distress data are used for pavement condition analysis and evaluation. The Thruway Distress Survey Manual (13) describes the process through which distress data are generated along the thruway pavement system. This process is labor-intensive, and represents one of the most involved data-gathering activities. Efficiency in the field effort has been improved through the use of laptop computers and the Distress Data Recording Computer Program (14). The collection of other field condition data (e.g., skid resistance, roughness, and structural deflection) is planned to be gradually introduced through fully or partially automated procedures.

The mass of data collected in the field is entered into the data base using a variety of ORACLE products, such as the SQL*Loader and PRO*C interfaces. The data flow diagram that describes the condition data collection process is shown schematically in Figure 3.

The collection of inventory (and historical) data represents an ongoing activity. Inventory data are obtained from a variety of sources, including as-built construction plans, standards, specifications, construction and material reports, traffic records, climate records, and others. As a substantial portion of inventory data is stored on paper (i.e., not in a machine-readable format), the effort of extracting the data values needed for the data base is cumbersome. Historical raw data, typically stored in manual files, are extracted and keyed in directly to predesigned data base tables.

Condition data are evaluated and analyzed using a variety of special-purpose computer programs. Some programs, such as the ones used for the determination of the pavement distress index (written in Pascal) and pavement damage analysis (15) (written in C with embedded SQL commands), have direct interfaces with the distress data that reside in ORACLE data tables.

Project-Level Analysis

The data base applications on project-level analysis (Figure 1) use six types of project-specific data. These data are in reference to records on maintenance history, climate, pavement condition, traffic, standard treatments, and contract pay items. The developed data flow diagram for project-level analysis is shown schematically in Figure 4.

Project-level analysis focuses on pavement segments having approximately uniform characteristics with respect to condition measures (distresses) and other factors (e.g., subbase and traffic). Condition data for the thruway system are collected along pavement segments having a length equal to 0.1 mi. This nominal pavement length also determines the structure in the data base used to store condition data. Condition, traffic, maintenance history, and other data are analyzed to
 establish the lengths of projects to be considered for maintenance and rehabilitation purposes.

Historical records on maintenance and rehabilitation are useful for understanding the causes of pavement deterioration and for evaluating pavement performance. Standard treatments, developed by thruway maintenance personnel over the years, cover a variety of options within the categories of routine maintenance, resurfacing, restoration, rehabilitation, and reconstruction. Pay items define the units of work and associated costs for pavement treatments implemented through contracts. Data elements include quantities and unit costs for construction materials, labor, and equipment for given specifications.

The treatment selection methodology is developed within a knowledge-based system being developed using the Nexpert Object as the expert system shell. The knowledge-based system can be interfaced with either the ORACLE database or the part of the PIDBS that resides in the ORACLE environment. In the latter case, data needed for knowledge-based applications can be transferred to a local data base and be used for the purposes of each specific application.

The final product of project-level analysis is a set of feasible treatment options for each pavement project. These options are stored in a separate data base table, and are used to select the optimal treatment through network-level analysis.

### Network-Level Analysis

Network-level analysis considers the relative conditions and competing needs of individual thruway sections and divisions in order to allocate resources along the entire thruway system. It uses project lengths and candidate treatments determined during project-level analysis in order to select and preferentially rate the most cost-effective maintenance and rehabilitation actions. These tasks of network analysis are implemented so as to achieve defined pavement condition and, at the same time, satisfy budget and other constraints.

The selection of the most cost-effective treatments from the pool of options is achieved using an optimization model. This model uses dynamic programming techniques to obtain minimum-cost strategies for each project length over a given analysis period. Required input includes pavement condition, age, treatment history, and costs associated with the treatment options. The output of the analysis provides the optimal path for each project length over the analysis period, expressed in terms of a projected sequence of treatments and related costs. A preferential rating scheme establishes the short- and long-term programs that can satisfy existing and projected budgets and other constraints.

The data flow diagram of the network level analysis is shown schematically in Figure 5. The computer program performing the optimization analysis is expected to be interfaced with the data base to easily access needed data and store results after processing.

### DATA MODELING

In general, many issues must be resolved before completing the functional design of a data base system (17). Examples of such issues include the selection of an appropriate data
structure for implementing the data base; definition of data properties, data representations, and others. This was also the case for the functional design of the PIDBS.

The starting point of the effort was to study a wealth of documents on pavement-related activities at the NYSTA and to conduct interviews with a wide range of users. This procedure helped obtain a better understanding of the nature and objectives of each function of the data base. During system analysis, all pavement management–related functions were decomposed into three components: namely data collection, project-level analysis, and network-level analysis. At the early stage of system analysis, the data stores included in each of the three components were not clearly identified. Using a data modeling technique (7), it became possible to provide a detailed analysis of each component of the system. The technique depicted the entity relationships within each component of the data base system, which, in turn, assisted to achieve an effective communication with users.

The developed E–R model has a semantic content, in which the real world is viewed as being composed of entities and relationships among these entities. An important characteristic of the E–R model is its ability to justify and explain the data structure and data models used in implementing the data base. The procedure used to design the data base using the E–R model (7) involves the following four tasks:

- Identification of the entity sets and their relationships;
- Definition of attributes, data types, and allowable values; and
- Organization of data into entities and relationships, and selection of primary keys.

The produced data base system includes 33 entities. Examples of included entities are accidents, contract pay items, contracts, contractors, divisions, climate, lane segments, pavement conditions, pavement maintenance history, political jurisdictions, roadway segments, sections, tests, traffic, standard treatments, and uniform sections.

There are three types of relationships between entities: namely one-to-one (1:1), one-to-many (1:n), and many-to-many (n:m). These are shown in Figure 6 for a relationship defined as “subject-to” between the two entities pavement conditions and standard treatment. Each entity is represented in the figure by a rectangular box, and each relationship is represented by a diamond-shaped box. It should be noted that, in general, this specific subject-to relationship is a one-to-many type; i.e., one pavement condition may be subject to many standard treatments. The specific type of each relationship depends on many factors, such as maintenance practices and existing policies. For instance, in the case of the subject-to relationship if more than one pavement condition could also be subject to one standard treatment, then the relationship would be many-to-many.

The results of the applied E–R model were used for the implementation of the PIDBS. A CASE tool was also used to provide graphic presentations of the established entity re-
Each condition may be subject to one and only one treatment.

Each condition may be subject to one or more treatments.

Each treatment may be subject to one or more treatments.

Each treatment may be applied to one or more conditions.

**FIGURE 6** A sample E-R diagram.

**DATA DICTIONARY**

The data dictionary is a repository of information about the definition, structure, and usage of data. It is generally regarded as a system, rather than a user, data base and is a significant tool to ensure consistency of all elements in the system. Its purpose is to provide better documentation, control, and management of the organization of data.

Functions provided by the data dictionary include the following:

- Definition of attributes and tables;
- Definition of relationships;
- Definition of information about tables accessible by a particular user, and definition of the type of access;
- Definition of data uses and sources;
- Definition of data views;
- Definition of input-output screens; and
- Definition of reports.

An example of an entity documented in the data dictionary is presented in Table 1. It is in reference to the project-level functions of the entity standard treatments. It is noted that each attribute value in the examples is specified in terms of data type (e.g., number, character, and date) and allowable value. The attributes and values defined for the entity as well as the primary keys (attributes prefixed with asterisks) can be also seen in the table.

**DATA BASE APPLICATIONS**

**Application Interfaces**

SQL implemented in the ORACLE RDBMS features both interactive and programmatic application interfaces. Interactive applications (e.g., report generation) use SQL as the primary query language. In programmatic applications (e.g., data analysis), however, SQL is considered as a sublanguage
with data manipulation commands placed in the application programs.

Examples of interactive and programmatic interfaces are as follows:

**INTERACTIVE:**

```sql
SELECT TCODE, TTYPE
FROM STD_TREAT
WHERE TREAT_EXPT_LIFE > = 10;
```

**PROGRAMMATIC:**

```sql
EXEC SQL SELECT TCODE, TTYPE
INTO :TCODE, :TTYPE
FROM STD_TREAT
WHERE TREAT_EXPT_LIFE > = 10;
```

In the example of programmatic interface, the prefix EXEC SQL is needed to distinguish the SQL commands from the C language statements that surround it. Likewise, the INTO clause is needed to designate the input area; and the variables named in that clause must have a colon prefix in order to be distinguished from data base field names. Applications using such programmatic interfaces possess the power of SQL and the flexibility of a procedural language.

### Report Generation

An interactive application commonly used in PIDBS is report generation. The functional specifications of the data base system document the procedures used for report generation. These procedures are as follows:

- Organization of standard reports produced from the data base,
• Characterization of reports into classes of applications, and
• Description of menus presented to end users for report production and data entry.

Specific reports produced for pavement management purposes are as follows:

• PDI reports, which provide distress condition summaries for each thruway section and division, or the entire thruway system;
• Condition history reports of pavement segments for project analysis;
• Field schedule reports for condition data gathering;
• Contract work reports for projects performed at each pavement segment;
• Budget planning reports for short- and long-term maintenance programs;
• Maintenance history reports, which summarize past maintenance and rehabilitation actions; and
• Traffic volume reports, which provide AADT, ADTT, and average truck load distribution factors for each candidate project.

The reports are created by entering search criteria and parameters into the SQL*ReportWriter through the fill-in-theform interface. This interface provides a list of report definition screens, namely Query, Group, Field, Summary, Text, Report, and Parameters. Entering the RUNREP command from the operating system, the user can execute previously defined reports in batch mode without intervention.

Reports use selection criteria for the elements to be included in the report; sorting criteria for placing desired elements in order; and grouping criteria for assigning elements to a defined group. Basically, all information stored in the data base that pertains to each element is included in the report, unless the user selects different criteria that would result in a customized report.

Programmatic Applications

In the ORACLE product, SQL statements can be embedded in a program written in COBOL, C, Fortran, or PASCAL language. The present work has used C as the host language for most of its application programs.

The PRO*C interface provides a tool that enables writing ORACLE data base application programs and indirectly accessing the pavement data base.

The programmatic interface applications that are being implemented in the pavement management system of the NYSTA are briefly described in the following paragraphs.

Damage Assessment

Pavement condition expressed in terms of distresses along the thruway pavement system is being evaluated annually. Condition data are being analyzed using a variety of techniques that aim to establish condition indices, deterioration rates, and other parameters of pavement performance.

Project-Level Analysis

Distress data are being merged with other data types to achieve a detailed diagnosis of pavement condition and analyze causes of deterioration. Results of project-level analysis include treatment alternatives and costs.

Network-Level Analysis

Treatment selection for each specific project is achieved by introducing a systems perspective to the project-level analysis with consideration of total costs, system goals, etc. It aims to allocate resources on the basis of the relative conditions and competing needs of the entire thruway pavement system.

Life Cycle Cost Analysis (LCCA)

A cost analysis methodology used engineering expertise and judgment in deciding the appropriateness as well as performance characteristics of treatments. Following the AASHTO guidelines, LCCA can be performed to determine, for each project, the best alternative in minimizing the overall cost over a period of time.

PROTOTYPE IMPLEMENTATION

Following the completion of the functional design of the system, a prototype version of PIDBS was produced. The prototype is considered as a developmental model of the system primarily for testing purposes. It has been reviewed by users, revised, and expanded to serve as a working model of the system.

The objectives of the prototype system were to (a) serve as a learning device, and (b) provide a working version of the system to be used in testing the interface capabilities of PIDBS.

Important criteria considered for the development of the prototype system included the following:

• Users expected to test the system possess a good understanding of the identified data needs;
• The data source in the prototype should be accurate, and be made available to users; and
• The size of the prototype should be small enough to be manageable, yet adequate to allow a wide range of users to evaluate its capabilities.

On the basis of these criteria and the established functional specification, the scope of work for creating the prototype system involved the following activities.

Data Flow Diagram

The entire pavement management process is illustrated with data flow diagrams in a top-down manner. This establishes a hierarchy of levels of detail for this complex system, and allows the system analysts to reduce each complex problem to clearly defined entities.
E-R Diagrams
E-R diagrams provide graphical representations of the data structure in achieving the objectives of data collection, project-level analysis, and network-level analysis. They possess the advantage of semantic content, which aids to achieve an effective communication with the users, and present the detailed analysis of the system.

Data Dictionary
The data dictionary defines and documents attributes, entities, relationships, data uses, data sources, tables, views, user access types, and other data-related operations. It is also used to coordinate the system development effort and to ensure consistency during the development process.

Automatic Data Conversion
Field condition data (recorded with laptop computers) and inventory data (stored in dBASE or computer-readable files) are being automatically transferred into ORACLE data base tables. This is achieved with the aid of PRO*C and SQL*Loader interface programs.

Screen Input Design
Inventory data not stored in a computer-readable format are typed into the data base. A number of screens are designed to facilitate such data entry.

Form Trigger Design
The SQL data manipulation language is used to pursue trigger design. This aims to enable users to view certain types of data which are sorted, grouped, or derived from raw data. Forms are designed so as to have the capability to trigger either from a single table or from multiple (joined) tables.

Report Generation
Reports are important for presenting information at all levels of, and stages in, decision making. Pavement condition reports at the project and network levels are among the most frequently used. These can be in text or graphical form.

Menu Design
The front end of the system is given by a menu-driven interface. The menu structure is designed in a top-down manner, and involves a hierarchical menu tree that allows users to access the data base in a variety of ways. These include access to another menu, a formatted screen, a customized report, or an application program. The main (top) menu provides several options (applications) that allow access to specific data entities.

Programmatic Applications
Numerous interface programs have been developed and are being interfaced with the data base. Two specific application interfaces implemented in the prototype are the pavement distress index (PDI) determination and the damage assessment methodology.

DISCUSSION OF RESULTS
The PIDBS is an important component of the pavement management system under implementation at the NYSTA. The PIDBS serves the needs for data storage and retrieval and provides an integration of the database system with application programs. Thus, the PIDBS aims to achieve the important objective of easy access to all types of information needed in the pavement management process.

The presented functional specifications of the PIDBS were produced by considering the data needs of end users and the requirements to provide a linking mechanism among the various applications in pavement management. In this sense, the functional specifications have established the pattern for both the database implementation and the design of application interfaces.

An examination of available database systems for pavement management purposes has found that such systems are typically agency dependent. Among the systems examined, some had insufficient data modeling techniques for the database design.

The development of the PIDBS has combined the approach of both system and data analysis. System analysis has been conducted to investigate the pavement management process, and has produced logical data flow diagrams to represent this process in a top-down manner. Data analysis has been conducted to synthesize data sources and E-R models, and followed a bottom-up approach.

A CASE tool was used to represent graphically the system structure, data flow diagrams, and E-R diagrams. Central to the development of the PIDBS was the design of the E-R model and the establishment of the requirement of data independence. This has the important advantage of allowing tuning of the physical database to enhance efficiency without having to re-write or discard application programs. The mapping from the E-R model to the relational model maintained data consistency and integrity, while it reduced data redundancy.

SUMMARY AND CONCLUSIONS
The work of formalizing the functional specifications of the PIDBS (an integrated database system for pavement management) has been described. The formalization consisted of describing the system structure, identifying entities and relationships, documenting data dictionary, organizing report format, and designing programmatic interfaces.

The functional design began with system analysis, which followed a top-down design and decomposed the system structure into functional components. The data stores of each component were identified with the E-R data modeling technique. The data structure of entities and relationships and the
Properties of data elements were defined and documented in the data dictionary. The query language SQL has been used in the application interfaces for data manipulation. The procedures of reports generation and the purposes of analysis programs were also presented in the form of specifications.

A prototype system has been developed in the ORACLE RDBMS with the aid of a CASE tool, and Pascal and C programming languages. Included are data flow diagrams, E–R diagrams, data dictionary, automated data conversion programs, input screens, forms trigger, reports generation, menus design, and application interfaces.

On the basis of the development to date, the following conclusions may be drawn:

- Functional specifications that clearly define the structure and capabilities of the data base system are an important requirement for database design.
- The functional design of a data base system requires a complete analysis of the structure of the system.
- The E–R model is suitable for the analysis of data in pavement management.
- The development of a prototype data base system involves a highly iterative process.
- The structure of a well-designed PIDBS should allow it to be easily integrated with other existing or future relational data bases.

ACKNOWLEDGMENT

This research has been supported under a contract between Rensselaer Polytechnic Institute and the NYSTA. It is part of a broader research effort to develop a pavement management system for the NYSTA. Input provided by NYSTA personnel during the system development is gratefully acknowledged.

REFERENCES


