Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
SWUTC/02/167511-1		
4. Title and Subtitle Developing an Integrated Management System for the Urban Transportation Infrastructure		5. Report Date February 2002
7. Author(s) Zhanmin Zhang , Fuat Aki, and W. Ronald Hudson		6. Performing Organization Code 8. Performing Organization Report No. Research Report 167511-1
9. Performing Organization Name and Address Center for Transportation Research University of Texas at Austin 3208 Red River, Suite 200 Austin, TX 78705-2650		10. Work Unit No. (TRAIS) 11. Contract or Grant No. 10727
12. Sponsoring Agency Name and Address Southwest Region University Transportation Center Texas Transportation Institute Texas A&M University System College Station, Texas 77843-3135		13. Type of Report and Period Covered 14. Sponsoring Agency Code
15. Supplementary Notes Supported by general revenues from the State of Texas.		

16. Abstract

This report describes the development of an integrated management system for urban transportation infrastructure. The developed system integrates pavements and bridges at both network and project level management.

A computer program was developed as a result of this study. The developed computer program is user-friendly and takes full advantage of Graphical User Interface (GUI) technology. In addition, a combined priority index was developed to compare pavements and bridges on the same scale. In developing this index, the variables that are common to both pavements and bridges were used including: 1) Traffic, 2) Condition, 3) Width, and 4) Age.

17. Key Words	18. Distribution Statement
Urban Transportation Infrastructure, Infrastructure Management System, Pavement and Bridge Management, Integrated Management System, Urban Roadway Management, Brigit, TBSS	No restrictions. This document is available to the public through the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161.

19. Security Classif. (of report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	150	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

DEVELOPING AN INTEGRATED MANAGEMENT SYSTEM FOR THE URBAN TRANSPORTATION INFRASTRUCTURE

by

Zhanmin Zhang
Fuat Aki
W. Ronald Hudson

Research Report SWUTC/02/167511-1

SOUTHWEST REGION UNIVERSITY TRANSPORTATION CENTER
Center for Transportation Research
University of Texas at Austin
Austin, Texas 78712

February 2002

DISCLAIMERS

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

ACKNOWLEDGMENTS

The authors recognize that support for this research was provided by a grant from the U.S. Department of Transportation, University Transportation Centers Program to the Southwest Region University Transportation Center which is funded 50 percent with general revenue funds from the State of Texas.

ABSTRACT

This report describes the development of an integrated management system for urban transportation infrastructure. The developed system integrates pavements and bridges at both network and project level management.

A computer program was developed as a result of this study. The developed computer program is user-friendly and takes full advantage of the Graphical User Interface (GUI) technology. In addition, a combined priority index was developed to compare pavements and bridges on the same scale. In developing this index, the variables that are common to both pavements and bridges were used including: 1) Traffic, 2) Condition, 3) Width, and 4) Age.

EXECUTIVE SUMMARY

1. INTRODUCTION:

The maintenance of the existing infrastructure is a vital and complicated issue that requires millions of dollars each year. An infrastructure management system is a tool to help decision makers maintain their infrastructure at an acceptable level of service for the public by means of systematic, and coordinated procedures [Hudson 96]. Such management systems have been used over the last 25 years for transportation infrastructure, specifically for pavements and bridges.

One of the main problems with the existing infrastructure management systems is that for each different type of infrastructure, a different individual management system has been used. For example, although pavements and bridges are both types of transportation infrastructure, there are separate management systems for the two. If the scope of the concept is broadened to manage all the infrastructure within a city in addition to the transportation infrastructure, the number of management systems required increases accordingly.

In fact, the problem is not the number of different management systems, but the lack of communication among such systems. It is useful for an individual agency to develop and implement a management system for a particular type of infrastructure. However, considering the interactions among different types of infrastructure and considering the efforts and money spent on each individual management system, this may not be the most economic overall solution for the public. For example, it is possible that a newly resurfaced pavement can be dug up to maintain or install utility ducts. In this example, both pavement and utility management systems may have done their jobs perfectly, they might even use complicated mathematical models to optimize the benefits of the agencies concerned. No matter how good the result of each system is individually, the overall result may be unfeasible because of improper interaction.

A similar problem can exist for pavements and bridges. It is not reasonable to make people suffer from a traffic delay due to pavement rehabilitation work, followed two months later by another delay due to bridge maintenance work on the same street. It would be more feasible to share information between these two management systems and schedule projects that would minimize total delay by performing these two actions simultaneously.

This report describes an Urban Integrated Management System (UIMS), which integrates pavement and bridge management systems into a single package. The software developed is a user-friendly program and takes full advantage of Graphical User Interface (GUI) technology. A combined priority index was developed to compare pavements and bridges on the same scale. In developing this index, the variables that are common to both pavements and bridges were used including: 1) Traffic, 2) Condition, 3) Width, and 4) Age.

2. CONCEPT OF INTEGRATED MANAGEMENT SYSTEMS:

Infrastructure management is a process of planning and programming investments or expenditures, which includes design, construction, maintenance, operation, and in-service evaluation of physical facilities [Hudson 96]. Infrastructure management systems have been under development since 1970's when the infrastructure crisis was identified. Many local agencies were asked to do more with less after 1970 when their budgets were reduced [Sachs 93]. This enhanced the idea of developing management systems that provide decision support for funding requests and deciding how limited funds should be allocated to maintenance and rehabilitation needs.

Pavement management systems were generally the earliest of the structured infrastructure management systems implemented by many agencies [Hudson 94, Sachs 93]. The early studies began in 1968 at the University of Texas. The objective of this study was to develop new pavement design methods using the systems approach [Hudson 94]. Since that time, a significant amount of research has been performed in this arena. Today, pavement management systems are being implemented all around the world.

Another area of structured infrastructure management is bridge management. The collapse of the Silver Bridge across the Ohio River in 1967 initiated the need for formalized concepts of management systems for bridge infrastructure [Siccardi 93].The collapse focused the attention of the Congress on bridges, and a program for rehabilitation and replacement of bridges began using the concepts of sufficiency rating, functional obsolescence and structural deficiency.

Cost is one of the important factors in developing and implementing a management system for any type of infrastructure. For an economically feasible management system, the benefits provided by the system should be higher than the cost of implementing the system itself.

These costs include the expenditures during the development procedure, expenses of training the personnel, and computer and technology costs.

In public works, there are several types of infrastructure to be managed. For each type of infrastructure to be managed, a separate management system is generally needed, such as pavement management systems and bridge management systems. This brings about problems for each individual management system, for example, how money should be spent for development and implementation purposes. Most of the time, at the executive level, these infrastructure facilities are operated by different agencies which have separate budgets and their objective is to keep the facility operating at its maximum level of service with the available funds. However, at the administrative level where the total budget is divided among these different types of infrastructure, it is the decision maker's responsibility to consider all possible effects and make the most cost effective decision.

In addition, even if a particular management system has the most sophisticated models and methodologies to find optimum solutions for a particular infrastructure type, the overall optimum is always affected by other factors. For example, utility lines are usually placed under the pavements; therefore, if effective management is to be achieved, the decision maker must take into consideration how these infrastructures interact with each other.

These problems and questions are addressed adequately in the concept of integrated management systems. In this concept, it is assumed that several types of infrastructure can be integrated within a single infrastructure management system which provides decision makers information to examine the impact of various alternative scenarios [Hudson 96]. In addition, the cost of implementing a single system is much less than the total cost of implementing several individual management systems.

The main philosophy behind integrated management systems is to take the common aspects of different management systems and combine them in a single system. The common aspects of management systems can be outlined as follows [Hudson 96]:

1) Analysis modules.
2) A central database.
3) Economic analysis models.
4) Optimization models.

5) A graphical interface (e.g. GIS).

The most important aspect among these is the database which includes the physical attribute and condition data for each type of infrastructure. The easiest way of integration is to combine these databases using some common data fields so that information sharing becomes possible between different infrastructures.

3. MODULES AND FORMULATIONS:

Formulation of the modules in the UIMS involves three parts: 1) Formulation of the pavement management module, 2) Formulation of the bridge management module, and 3) Formulation of the integrated system.

3.1 Formulation of the pavement management module: In UIMS, the procedure for pavement management is a one-year analysis procedure, which addresses solutions for the current deficiencies. The Decision Tree Method is used to assign the maintenance and rehabilitation (M&R) strategy to pavements. The Decision Tree Method is a very popular method where the variables used in the decision-making process are taken into account step by step.

The number and type of pavement M&R strategies in pavement management vary from agency to agency. The most popular M&R strategies include; do nothing, routine maintenance (repair distresses), thin overlay, thick overlay, and reconstruction. Three decision variables: 1) Pavement condition index (PCI); 2) Pavement age (AGE); and 3) Average daily traffic (ADT), are used in M&R strategy selection for pavements. Each variable is divided into five levels as shown in Table 1.

Table 1. Classification of Decision Variables

Levels	*PCI*	AGE	**ADT**
1	Bad	Very Old	Very Heavy
2	Poor	Old	Heavy
3	Fair	Average	Medium
4	Good	New	Light
5	Excellent	Very New	Very Light

Using these three decision variables with five levels for each, result in a total of 125 (5x5x5=125) M&R alternatives. Such a large number of alternatives are not very practical in implementation. The total number of M&R strategies is therefore reduced to a reasonable level. Because most cities use 10 to 20 M&R strategies for budgeting and programming, the first and last two levels of variables are combined together in one level as shown in Figure 1.

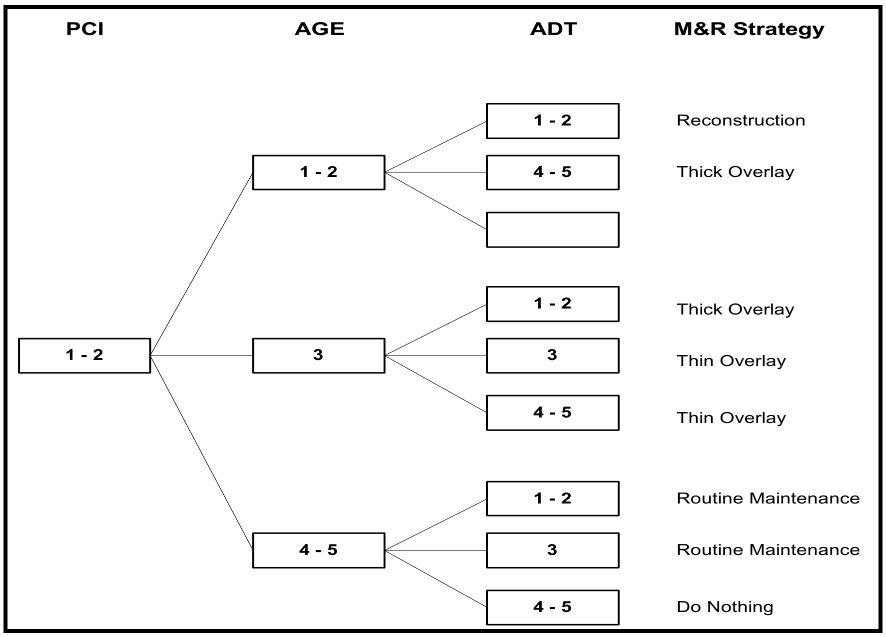


Figure 1. One Branch of the Decision Tree

3.2 Formulation of the bridge management module: The bridge management module consists of three parts:

1. A flagging mechanism for identifying the bridges with certain deficiencies in order to schedule special inspections, or assign operational restrictions.
2. A decision tree for assigning specific M&R projects for the three main bridge elements: deck, substructure, and superstructure.
3. A decision tree for assigning operational restrictions (like bridge posting) and scheduling special inspections.

The formulation takes full advantage of the NBIS data items that are collected every two years nationwide. Since many cities do not maintain any inventory for their bridges (cities take the inventory data from the NBIS data which is collected by DOT's), the formulation does not include a detailed inventorying module. It is important to note that several safety data items are emphasized through the flagging mechanism and some basic bridge identification data are included for proper reporting.

The bridge management formulation is a simple filtering mechanism. Table 2 lists the variables used in flagging filtering. Each variable has two levels.

Table 2. Classification of Decision Variables for Flagging Filter

	Level 1	***Level 2***
Historical Significance	Significant	Non-Significant
Bridge Length	Long	Short
ADT	Heavy	Light
Condition	Poor	Good
Safety Features	O.K.	Not O.K.
Scour Critical	Critical	Non-Critical
Deck Width	Narrow	Wide
Load Restriction	Restricted	Non-Restricted

According to Table 2, for example, the user can get a list of scour critical bridges, bridges with load restrictions, narrow bridges, etc. This list can also be used in further analyses. Bridge management formulation is also based on the present condition approach as in the pavement management formulation. The solution is reached by means of decision trees. The first decision tree is for assigning operational restrictions and scheduling special inspections. Four variables are used in this decision tree as shown in Table 3.

Table 3. Classification of Decision Variables for First Decision Tree

Levels	Scour Code	Deck Width	Load Restriction	Safety Feature
1	Very Critical	Large	Restricted	Not O.K.
2	Critical	Narrow	Non-restricted	O.K.
3	Non-critical			

Decision tree structure for assigning operational restrictions and scheduling special inspections is shown in Figure 2. It should be noted that the activities shown in this figure are examples of possible activities. In a real system, the user has the flexibility to set any kind of activity that is applicable to a particular agency.

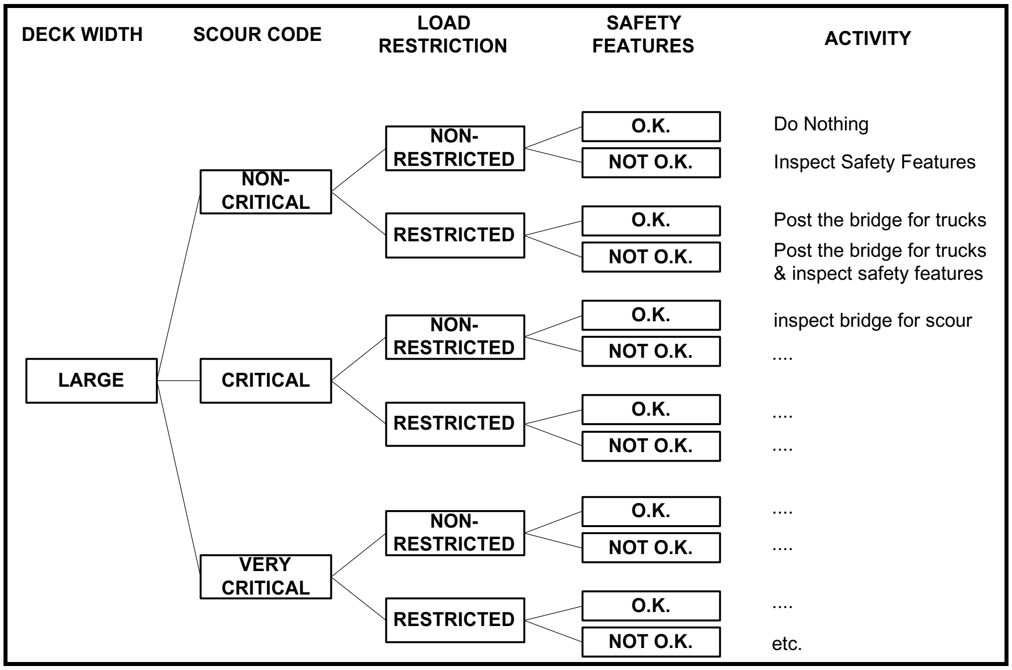


Figure 2. One Branch of the Decision Tree for Operational Restrictions and

Special Inspections

Three main elements of bridges are used in assigning specific M&R project assignments: deck, substructure and superstructure. The M&R decision tree uses the condition values of these three elements as the decision variables. There are three levels for each variable as shown in Table 4.

Table 4. Classification of Decision Variables for M&R Decision Tree

Levels	Deck Condition	Substructure Condition	Superstructure Condition
1	Bad	Bad	Bad
2	Marginal	Marginal	Marginal
3	Good	Good	Good

3.3 Formulation of the Integrated System: In order to evaluate bridges and pavements simultaneously, a common priority index is developed. In developing this index, the evaluation criteria used in pavement and bridge management systems are examined to see if they can be combined. The objectives of a bridge management system are examined to see if they can be combined. The objectives of a bridge management system can be generalized to all types of infrastructure and can be used in developing a combined index for bridges and pavements. Figure 3 depicts a diagram that shows the objectives of an infrastructure management system and associated attributes to achieve these objectives.

In this diagram, the major attributes for achieving the objectives are depicted on the last line, including the width, condition, ADT and cost per vehicle.

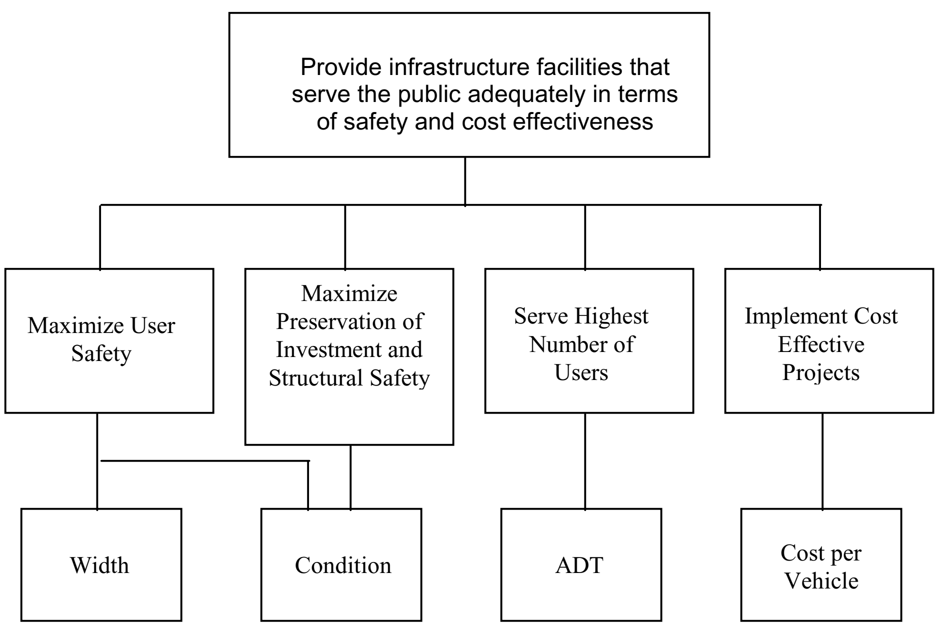


Figure 3. Objectives and Associated Attributes [Weissmann 90]

With some modifications in the bridge and pavement attributes, a combined evaluation index is developed for both bridges and pavement attributes, a combined evaluation index is developed for both bridges and pavements. In the combined index ADT, age, width and condition attributes are used. There is a need to modify the condition attributes of bridges so that they can be compatible with the pavement condition index. The condition of a bridge is represented by the minimum conditions of deck, substructure and superstructure. Bridge conditions are evaluated with numbers between 0 and 9 whereas the pavement condition index is evaluated between 0 and 100. In order to evaluate the two attributes on a common scale, the bridge condition values are multiplied by 100/9 (or 11.1).

Age is the difference between the current year and the year of construction for a bridge; or the difference between the current year and the year of the last major rehabilitation action or construction for a pavement section. ADT is the daily mixed traffic value, and width is the width of a bridge deck or pavement section. Each attribute is represented in three levels.

Two matrices are used in the calculation of the priority index as shown in Figure 4. The priority index is the summation of the two corresponding values coming from each matrix. Figure 4 also shows an example for the calculation of priority index (PI), for condition level of 3, age level of 2, ADT level of 2, and width level of 2, the priority index is 13. Small PI means that the structure should be given high priority when considering M&R project assignments.

Condition Levels	AGE Levels 1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

ADT Levels	Width Levels 1	2	3
1	1	2	3
2	4	5	6
3	7	8	9

PI = 8 + 5 = 13

Figure 4. Priority Index Matrices

The levels for variables are defined as shown in Table 5.

Table 5. Levels of Variables for Prioritization Procedure

Levels	Condition	Age	ADT	Width
1	Bad	Old	Heavy	Narrow
2	Fair	Medium	Medium	Medium
3	Good	New	Light	Wide

4. SYSTEM DESIGN:

The key factor in an effective infrastructure management system is a relational, centralized database. All the modules and functions within the system communicate with this database to perform the analyses by means of a computerized system. Therefore there are two important parts in system design:

1. Database design, which includes the design of database tables and their relations; and,
2. Overall system design, which includes the data flow diagrams through the different models of the system.

4.1 Database Design: Database design includes defining the data-items that should be included in the database tables, which are used to store and manipulate the data. There are two types of database tables in the system:

1. Raw data tables that include the inventory and condition data related to the infrastructures, as well as any user inputted data. This is the type of data that is collected in the field to be put into the system as the initial input. One of the major functions of an infrastructure management system is the storage and maintenance of the inventory data. Raw data tables provide functions like entering, modifying or viewing the inventory data. They also provide input for the analysis modules of infrastructure management systems.
2. Processed data tables that contain the results from all calculations in the system. The working principle of the system is fairly straightforward: the setup information, raw inventory, and condition data are combined together with the analyses formulations to perform the analyses. During these procedures, the processed data tables are used to store the combined and calculated data.

The level of data detail should be well defined when setting up an infrastructure management system. If the purpose is to answer all possible questions using the management system, then the data should be as detailed as possible. However, the problem of collecting, inventorying and managing these detailed data may become very difficult and the system may be unfeasible for implementation due to the excessive data requirements. When designing an infrastructure management system, efforts should be given to the use of the existing data. A bridge management system tool like Pontis [Pontis 91] requires so many more data items than what is available on the National Bridge Inventorying System (NBIS) that it is almost economically unfeasible for many agencies to implement it.

Since this study includes two types of infrastructure, there are two sets of database tables: one for pavements and the other for bridges. The data items used are those which are currently available. For example, the NBIS database, which is available for each transportation department and updated every two years, is used as the bridge data source.

4.2 Overall System Design

System design defines the flow of information through the database tables. Generally, the physical attribute and condition data must be provided by the user via inventory and condition tables. The decision tree structures, critical values for variables to be used in decision trees, M&R actions and their unit costs are defined in the setup tables. Proper setup information is combined together with the physical attribute and condition data to perform the analyses. The overall system structure is depicted with the diagram in Figure 5.

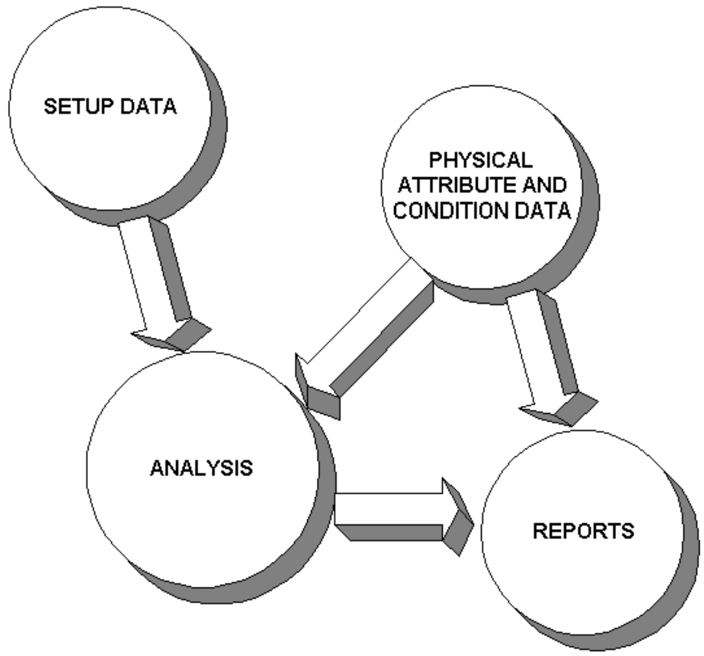


Figure 5. Overall Flow of Information

5. UIMS SOFTWARE:

The UIMS software is composed of four parts as shown in Figure 6: Setup, Database, Analysis, and Reports. The direction of the arrows in Figure 6 shows the direction of information flow within UIMS. First, the user defines the maintenance, rehabilitation and replacement (MRR) needs and decision tree structures that are used in analyzing the bridge and pavement infrastructure. This information is combined with the information in the database to perform the analyses. UIMS can produce reports for both the results of these analyses and the raw data itself.

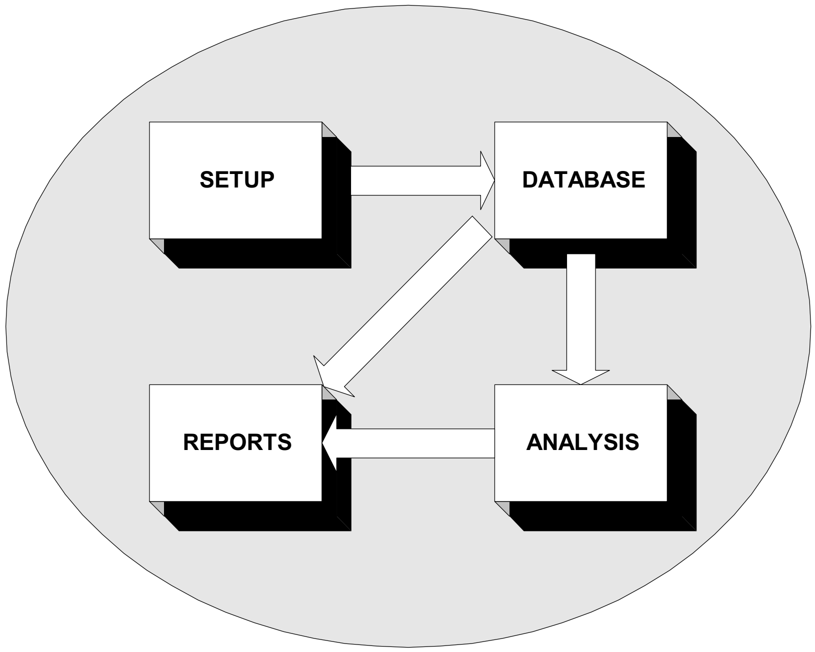


Figure 6. System Components of UIMS

UIMS is supported by a graphical user interface. The software is coded using PowerBuilder©, a powerful software package that can create applications that deal with small to medium-sized databases. The final product of UIMS is an executable computer program running under the Windows environment.

When UIMS is first started, the start-up window appears while the software connects to the database. This window displays only for a couple of seconds and disappears. After this point, UIMS is available to be used with the main window open. By means of the menus, the user selects the portion of the program to use. After each selection a new window shows up in the main window.

5.1 Utilities Menu: The first menu item in the UIMS is “Utilities”. This menu item includes the tools for setting up the general software requirements, including user sign-on and printer setup.

5.2 Setup Menu: The “Setup” menu is the second menu in the UIMS program and it provides the definition of the necessary information related to analyses. It contains two main parts, one for pavements and one for bridges. The setup menu is for the user to define maintenance and rehabilitation actions, the boundaries that separate different levels of different variables (for example, definition of the boundaries for low, medium, and high levels of ADT), and the decision trees that will be used in the analyses.

In the "***Evaluation Setup***" window, the user enters the lower and upper boundaries for the five levels of pavement condition index, age, and traffic. These values are then used in the analyses part when assigning the MRR projects to the management sections by means of the decision tree. In order to have an understanding about the overall values in the network, the user can press the "View Statistics" button. This button displays the pavement network statistical values.

5.3 Database Menu: The database menu is the third menu item in the UIMS main window. It includes the commands for basic data entry and data manipulation functions for both pavements and bridges. The first command in the pavement management part of the "Database" menu is called "***Inventory Data***". This command activates the inventory window for pavements or bridges, which provides functions like editing, viewing, or deleting inventory data. There are two tab buttons for switching between the "Free Form" and "Table" data window formats. "Free Form" format is good for entering new data into the database, because the user can see all the data items in one window. The "Table" form is good for data manipulations such as sorting and filtering.

UIMS provides functions for importing and exporting data, which are convenient for data backup purposes. It also saves considerable time on data entry if there are data available from an external data source. The "Import" button allows the user to import data from text or dBase files.

UIMS provides an extended querying mechanism to the user. This is performed via the "Filter" utility that can be used to perform manipulations with the current data. Using this button, the user can define various kinds of expressions as the filtering criteria. On the "Specify Filter" window, the user can select different functions from the function list and can select the column to be used in filtering from the column list. After setting a filter expression, the user can press the "Verify" button to check if the defined filter is appropriate or not. Logical operators such as "and" and "or" can also be used in defining the filter criteria.

Another utility under the database menu is called "***Statistics***". This utility activates the "Statistics" window that provides basic statistical information about the pavements or bridges. There are three parts in this window: a data window that displays the average values of ADT, AGE, and PCI; a graph selection part; and, a graph that displays the distribution for the selected variable.

UIMS allows the user to see the pavements and bridges on a map by means of the "***Street Map***" command of the Database menu. This command activates the street map window where the street map is plotted using the pavement and bridge inventory data. This window contains three main parts, the street map display window, the pavement inventory table, and the bridge inventory table.

5.4 Analysis Menu: The analysis menu is the fourth menu item in the UIMS main window and it includes the commands that provide pavement and bridge management analyses such as specific project selection and budget allocation. The windows under the analysis menu combine setup, inventory and condition data to analyze the infrastructure network. There are two parts under the analysis menu: Pavement Management Analysis and Bridge Management Analysis.

There are two tables on the "Pavement MRR Analysis" window, one is the pavement inventory table (at the top) and the other is the pavement analysis results table. When the user clicks on the "Apply Decision Tree" button, UIMS uses the decision tree setup information and pavement inventory data to select the MRR projects for each pavement management section. "Filter" and "Sort" buttons can be used if the user wants to introduce a filter prior to the selection or wants to rank the selected projects with a user defined criteria. For example, if the user wants to evaluate the pavement sections with 35 ft or less pavement width, a filter expression can be written using the "Filter" button. After the filter is set, UIMS updates the analysis results table for the filtered sections.

In the analysis summary window, the section code, the MRR action to be applied, the unit cost of the MRR action, project cost, and the total cost values are displayed. To distribute the available budget (which is usually less than the total cost), the "Calculate" button can be used. The user enters the available budget into the budget field and presses the "Calculate" button. This distributes the available budget among the projects starting from the top of the list until it finishes. For this reason, a proper sorting should be provided prior to the budget allocation.

The structure of the bridge management analysis part is different from that of pavement management analysis. Specifically, it includes a flagging analysis, a bridge MRR analysis and a special inspection analysis. The "Flagging Filter" window includes two tables, the inventory table (at top) and the flagging results table. There are also several radio buttons for selecting the filtering variable.

The "Special Inspection Scheduling" command used to select the bridges that require special inspection or operational restrictions. It is similar to the MRR project assignment window, the only difference being that instead of calculating cost, the next inspection date is determined by the software.

5.5 Reports Menu: The reports menu consists of a single command that activates the report selection window. UIMS allows the user to generate predefined reports from the analyses performed. Once the report selection window is opened, the user can view or print a report simply by clicking on the name of the report and pressing the "Show" button.

The "Show" button activates the report-viewing window. For example, if the user selects the "Report for Bridge MRR Results" from the report selection window, this report can be viewed in the report-viewing window. The user can zoom in and out of the report and print it using the "Print" button.

6. CONCLUSIONS:

An integrated management system (UIMS) for managing urban transportation infrastructure was developed in this study. UIMS is a computerized tool, which is intended to assist decision makers in managing their infrastructure effectively. The system consists of two subsystems; pavement management and bridge management. UIMS integrates these two separate systems by means of ultimate data sharing and also provides a single evaluation index to compare pavements and bridges simultaneously.

UIMS basically works in two ways: the user either can use it as a tool to manage pavements or bridges separately (like two separate programs), or make an integrated analysis to make more cost effective budget allocations and M&R project assignments. The UIMS software is a Windows-based application. It is a very user-friendly program supported with a graphical user interface (GUI) technology.

UIMS is a very flexible program, which makes it easier to adapt to different local conditions and requirements according to different agencies' needs. Most of the models used in the analyses are generic, and can be changed or modified by the user. UIMS filtering and sorting functions allow the user to set any kind of expressions when selecting or prioritizing M&R candidates; in other words, it does not restrict the user in utilizing a pre-defined priority index or a structural rating number.

The UIMS database is a relational database, which contains inventory and condition data for both pavements and bridges. The software provides import and export options that make it faster in cases where data is available from external sources. UIMS can import text (ASCII) and dBase (DBF) files, which can be generated by most of the other computerized systems. The export option provides more alternatives to save the data into a specific file format, which allows the user to make some other analyses using some other packages such as spreadsheet files like Excel. The main purpose of exporting is to provide a backup mechanism for the system.

The data items used in the bridge management part of the UIMS are NBIS data items. This makes it easier to implement UIMS compared to other bridge management tools, because one of the major problems in implementing a bridge management system is the compatibility with the NBIS database. The import option of the UIMS provides direct reading of data from the NBIS database, thereby reducing the time of data entry.

UIMS is an effective tool in network level infrastructure management. It first provides some basic statistical values for the variables that are used in the analyses including ADT, length, width, average bridge condition, and pavement condition index (PCI). These statistics are used to evaluate the pavement and bridge networks in general. The user then decides on the levels of these variables and sets the decision trees that will be used in selecting M&R actions. UIMS then applies these decision trees and assigns M&R projects to pavements and bridges. The cost of each project is calculated and compared with a user-defined budget.

There are three different analysis options for bridges in UIMS: a flagging procedure that pops out the bridges with certain deficiencies such as scour critical and historical bridges, a decision tree procedure that assigns M&R projects to bridges, and another decision tree that assigns operational restrictions and/or special inspections. The special inspections procedure assigns an inspection date for the bridges that requires inspection due to an operational deficiency.

The street map function in UIMS provides the user to visually see the pavements and bridges on the screen. This function provides a quick reference for the user to identify the structures according to their different attributes such as low condition and high traffic. UIMS uses color-coding technology for this purpose.

UIMS provides many different alternatives for reporting purposes. The pre-defined reports that are accessed through the reporting module are readily available for the user to review and print reports. In addition, each window in the UIMS has a printing option that makes it possible to produce any kind of user-defined reports.

TABLE OF CONTENTS

Acknowledgements ...... **vi**
Abstract ...... **vii**
Executive Summary ...... **ix**
List of Figures ...... **xxix**
List of Tables ...... **xxxi**

CHAPTER 1. INTRODUCTION **1**
1.1 Objectives of the Study 2
1.2 Research Approach 3
1.3 Report Organization 4

CHAPTER 2. CONCEPT OF INTEGRATED MANAGEMENT SYSTEMS **7**

CHAPTER 3. CURRENT PRACTICES IN PAVEMENT AND BRIDGE MANAGEMENT **11**
3.1 A Network Level Bridge Management System: Pontis 11
3.1.1 Structure of Pontis 11
3.1.2 Problems in Implementation of Pontis 14
3.1.3 System Requirements for Pontis 14
3.2 Bridgit 14
3.2.1 Implementation of Bridgit 15
3.2.2 System Requirements for Bridgit 15
3.3 TBSS 15
3.3.1 Decision Variables Used in TBSS Formulation 17
3.3.2 Implementation of TBSS 17
3.4 Visual BMS 18
3.4.1 Visual BMS Overview 18
3.5 Urban Roadway Management System (URMS) 19
3.5.1 URMS Subsystems 19
3.5.2 Implementation of URMS 20
3.6 Visual IMS 21
3.6.1 Operation of Visual IMS 21
3.6.2 Implementation of Visual IMS 24

CHAPTER 4. MODULES AND FORMULATION **25**
4.1 Formulation of the Pavement Management Module 25
4.2 Formulation of the Bridge Management Module 28
4.2.1 Flagging Mechanism 29
4.2.2 Decision Tree for Assigning Operational Restrictions and Special Inspections 33
4.2.3 Decision Tree for Assigning Specific M&R Projects 34
4.3 Formulation of the Integrated System 36

CHAPTER 5. SYSTEM DESIGN 41
5.1 Database Design 41
5.1.1 Raw Data Tables 41
5.1.2 Calculated Data Tables 42
5.2 Overall System Design 42

CHAPTER 6 UIMS SOFTWARE 47
6.1 System Components 47
6.2 Menu Structure 48
6.3 Using UIMS 50
6.3.1 Window Structures and Functions 51
6.3.1.1 Getting Started 51
6.3.1.2 Utilities Menu 51
6.3.1.3 Setup Menu 55
6.3.1.4 Database Menu 61
6.3.1.5 Analysis Menu 72
6.3.1.5.1 Pavement Management Analysis 72
6.3.1.5.2 Bridge Management Analysis 74
6.3.1.6 Reports Menu 76

CHAPTER 7. CONCLUSIONS AND RECOMMENDATIONS 79
7.1 Recommendations for Future Directions 81

REFERENCES 85

APPENDIX A DATABASE TABLES 89

APPENDIX B EXAMPLE SOURCE CODES FOR UIMS SOFTWARE 103

LIST OF FIGURES

Figure 1.1 Research Approach ... 4
Figure 2.1 Concept of Integrated Management System ... 10
Figure 3.1 The TBSS Flow Chart ... 16
Figure 3.2 Overall Data Flow Diagram of URMS ... 20
Figure 3.3 Analytical Functions in Visual IMS ... 22
Figure 4.1 Decision Tree Method ... 26
Figure 4.2 One Branch of the Decision Tree ... 27
Figure 4.3 Pavement Management Formulation ... 28
Figure 4.4 Flagging Filtering Mechanism ... 33
Figure 4.5 One Branch of the Decision Tree for Operational Restrictions and Special Inspections ... 34
Figure 4.6 One Branch of the Decision Tree for M&R Project Selection ... 35
Figure 4.7 Bridge Management Formulation ... 37
Figure 4.8 Objectives and Associated Attributes ... 38
Figure 4.9 Priority Index Matrices ... 39
Figure 5.1 Overall Flow of Information ... 43
Figure 5.2 Data Flow for Pavement Management Analysis ... 44
Figure 5.3 Overall Diagram for Bridge Management ... 45
Figure 5.4 Data Flow for Bridge Management Analysis ... 46
Figure 6.1 System Components of UIMS ... 48
Figure 6.2 Main Menu Structure of UIMS ... 49
Figure 6.3 UIMS Main Window with Opening Logo ... 52
Figure 6.4 “About UIMS” Window ... 52
Figure 6.5 “Data Flow” Window With One Pop-up Window Open ... 53
Figure 6.6 User Information Window ... 54
Figure 6.7 Printer Setup Window ... 54
Figure 68 Changing Printer Properties ... 55
Figure 6.9 MRR Action Definition Window for Pavement Management ... 56
Figure 6.10 Decision Tree Setup for Pavement Evaluation ... 57

Figure 6.11 Distress Weight Setup 58
Figure 6.12 Decision Tree Structure for Pavements 59
Figure 6.13 Changing MRR Action for one Branch of the Decision Tree 60
Figure 6.14 Special Inspection Definition Window....................61
Figure 6.15 Flagging (Quick) Filter Values 62
Figure 6.16 Inventory Data Window for Pavements....................63
Figure 6.17 "Table" Format for Inventory Data Window 64
Figure 6.18 "Delete Row" Message Box 64
Figure 6.19 Select Import File Window....................65
Figure 6.20 Save As Window 66
Figure 6.21 Specify Filter Window....................67
Figure 6.22 Result of Filter for Pavement Inventory Data....................68
Figure 6.23 Specify Sort Columns Window 68
Figure 6.24 Pavement Distress Data Window 69
Figure 6.25 Pavement Statistics Window (for AGE)....................70
Figure 6.26 Street Map Window (with pavements only) 71
Figure 6.27 Street Map Window With Zooming 72
Figure 6.28 Pavement MRR Analysis Results....................73
Figure 6.29 Flagging Filter Window....................74
Figure 6.30 MRR Project Selection for Bridges 75
Figure 6.31 Special Inspection Scheduling Window for Bridges 76
Figure 6.32 Report Selection Window....................76
Figure 6.33 Report for Bridge MRR Results 77

LIST OF TABLES

Table 2.1 Benefits of Implementing A Bridge Management System 8
Table 4.1 Classification of Decision Variables 26
Table 4.2 Classification of Decision Variables for Flagging Filter 32
Table 4.3 Classification of Decision Variables for First Decision Tree 33
Table 4.4 Classification of Decision Variables for M&R Decision Tree 34
Table 4.5 Levels of Variables for Prioritization Procedure 39
Table 6.1 Command Buttons and Their Functions 50

CHAPTER 1
INTRODUCTION

The maintenance of the existing infrastructure is a vital and complicated issue that requires millions of dollars each year. An infrastructure management system is a tool to help decision makers maintain their infrastructure at an acceptable level of service for the public by means of systematic, and coordinated procedures [Hudson 96]. Such management systems have been used over the last 25 years for transportation infrastructure, specifically for pavements and bridges.

One of the main problems with the existing infrastructure management systems is that for each different type of infrastructure, a different individual management system has been used. For example, although pavements and bridges are both types of transportation infrastructure, there are separate management systems for the two. If the scope of concept is broadened to manage all infrastructure within a city in addition to the transportation infrastructure, the number of management systems required increases accordingly.

In fact, the problem is not the number of different management systems, but the lack of communication among such systems. It is useful for an individual agency to develop and implement a management system for a particular type of infrastructure. However, considering the interactions among different types of infrastructure and considering the efforts and money spent on each individual management system, this may not be the most economic overall solution for the public. For example, it is possible that a newly resurfaced pavement can be dug up to maintain or install utility ducts. In this example, both pavement and utility management systems may have done their jobs perfectly, they even might use complicated mathematical models to optimize the benefits of the agencies concerned. No matter how good the result of each system is individually, the overall result may be unfeasible because of improper interaction.

A similar problem can exist for pavements and bridges. It is not reasonable to make people suffer from the traffic delay due to pavement rehabilitation work, followed two months later by another delay due to bridge maintenance work on the same street. It would be more feasible to share information between these two management systems and schedule projects that would minimize total delay by performing these two actions simultaneously.

The concept of integrated management systems is based on the assumption that two or more similar types of facilities can be integrated into one system to produce more effective and efficient infrastructure management. The main philosophy behind integrated management systems is to obtain information sharing among various management systems using the common aspects of these systems. The common aspects of management systems can be outlined as follows [Hudson 96]:

1. A central database which is the heart of any good management system.
2. Analysis modules including the optimization models and economic analysis models.
3. A graphical interface supported with a geographical information system (GIS).

In this respect, there are two basic keys in developing integrated management systems and at least one of them should be satisfied in order to be able to setup a good integrated infrastructure management system:

1. Data Sharing: Each individual management system has its own database to store and manipulate data. Integrated management systems are based on the principle of sharing information among these individual databases rather then trying to create new databases which involve new data items or all the data items of each individual system.
2. Developing new analysis indices for an integrated system: It is important to have common evaluation indices for different types of infrastructures in order to be able to compare and evaluate them to select the best economic alternative.

1.1 OBJECTIVES OF THE STUDY

The objective of this study was to develop an integrated management system for urban transportation infrastructure, both at the network level and project level. The scope is the urban areas, especially small to medium-sized cities.

Two main transportation infrastructures are integrated in this prototype system: pavements and bridges, but others can be added. The developed system is a computerized tool that assists decision makers in managing their pavements and bridges, either individually, or combined. That is, the developed system can be used for pavement management or bridge management, or a combination of the two.

The major objectives of the study can be summarized as follows:

1. Develop a procedure for identifying and selecting maintenance and rehabilitation (M&R) projects for city pavements.
2. Develop a procedure for identifying and selecting M&R projects for city bridges.
3. Develop a single index that can be used to evaluate pavements and bridges simultaneously.
4. Develop a computer program which has graphical user interface technology to combine the mentioned objectives.

1.2 RESEARCH APPROACH

The research approach followed in this study can be summarized in four main categories as shown in Figure 1.1:

1. A literature review which includes the evaluation of pavement and bridge management systems in general, benefits of implementing integrated management systems, and several pavement and bridge management practices in order to be able to make a detailed problem definition and to define an overall methodology.
2. Conceptual design of the models in the new system, formulations for each model, determining the indices to be used, data requirements, conceptual design of the database, and conceptual design of the new computer program.
3. Programming of the new software according to the conceptual design, coding of the models and formulations, setting up the database, creating database tables and establishing relationships among them.
4. Documentation of all the studies to produce the final report of the project.

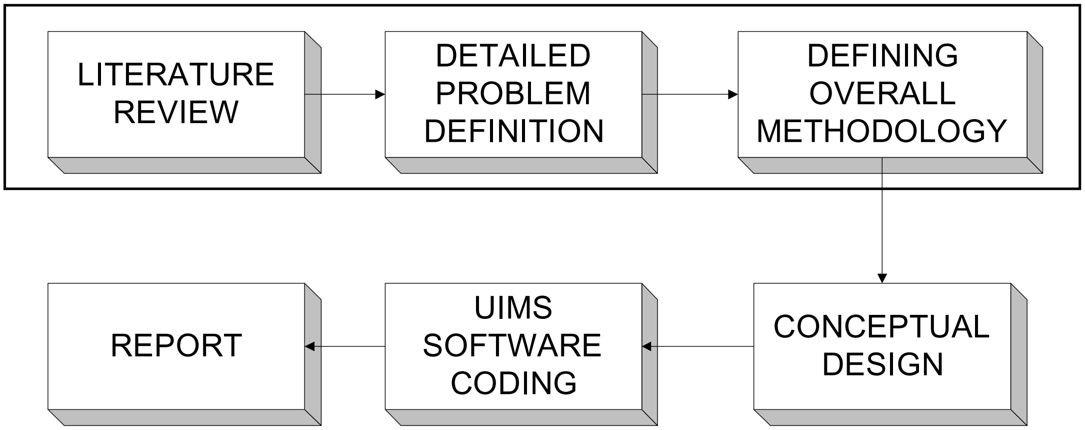


Figure 1.1 Research Approach

1.3 REPORT ORGANIZATION

This report consists of seven chapters. Chapter 1 provides a brief introduction including the problem definition and objectives of the study.

Chapter 2 summarizes a literature review and discusses the concept of management systems in general, especially the integrated management systems. Common methodologies used in pavement and bridge management systems are outlined, and benefits of implementing integrated management systems are emphasized.

Chapter 3 summarizes the literature review about the existing pavement and bridge management system practices, namely: Pontis, Bridgit, URMS, TEBBS, Visual BMS, and Visual IMS. This Chapter analyzes the different aspects of these systems including data requirements, ease in implementation, user friendliness, models and formulations, etc.

Chapter 4 discusses the models in the new integrated management system, and formulations for these models. There are two main models in the system: the pavement management model, and the bridge management model. Each of these models has several sub-modules, such as database, setup, and analysis. This Chapter discusses how the pavement management and bridge management models and their related sub-modules are formulated in this study. The formulation for the common priority index that is used to evaluate pavements and bridges together is also provided in this Chapter.

Chapter 5 describes the computerized system design for the new integrated management system, including the design of a database, the main system structural design, the data tables to be used in the system, the inter-relationships of these data tables (i.e. the flow of information between data tables), and the data items to be included in these data tables. This Chapter may be considered as a technical documentation for the UIMS software.

Chapter 6 describes the software developed in this study. It provides guidelines for “How to use the UIMS software” and can serve as a user manual for the UIMS software.

Chapter 7 is the concluding chapter and discusses the future research that can be performed on this subject, including the implementation of the new system.

CHAPTER 2

CONCEPT OF INTEGRATED MANAGEMENT SYSTEMS

Infrastructure management is a process of planning and programming investments or expenditures, which includes design, construction, maintenance, operation, and in-service evaluation of physical facilities [Hudson 96]. Infrastructure management systems have been under development since 1970's when the infrastructure crisis was identified. Many local agencies were asked to do more with less after 1970 when their budgets were reduced [Sachs 93]. This enhanced the idea of management systems that provide decision support for funding requests and deciding how limited funds should be allocated to maintenance and rehabilitation needs.

Pavement management systems were generally the earliest of the structured infrastructure management systems implemented by many agencies [Hudson 94, Sachs 93]. The early studies began in 1968 at the University of Texas at Austin. The objective of this study was to develop new pavement design methods using the systems approach [Hudson 94]. Since that time, more research has been performed in this arena. Today, pavement management systems are being implemented all around the world.

Another area of structured infrastructure management is bridge management. The collapse of the Silver Bridge across the Ohio River in 1967 initiated the need for formalized concepts of management systems for bridge infrastructure [Siccardi 93]. The collapse focused the attention of the Congress on bridges, and a program for rehabilitation and replacement of bridges began using the concepts of sufficiency rating, functional obsolescence and structural deficiency.

Implementing infrastructure management systems provide many benefits. For example, Table 2.1 shows the benefits of implementing a bridge management system at the three levels of an agency [Hudson 87].

Table 2.1 Benefits of Implementing A Bridge Management System

At Administrative Level	At Executive Level	**At Technical Level**
1) Summarizes bridge structural conditions 2) Summarizes bridge functional conditions 3) Addresses fund allocation questions 4) Establishes needs 5) Assists with statewide budget estimates	1) Prioritizes candidate projects 2) Analyzes cost effectiveness of various programs 3) Identifies bridges for posting 4) Prioritizes MR&R program, and tracks and schedules MR&R actions	1) Makes information readily available 2) Allows easy editing and manipulation of condition data 3) Provides details for project level design 4) Provides current costs 5) Allows easy special sorting and reporting

Cost is one of the important factors in developing and implementing a management system for any type of infrastructure. For an economically feasible management system, the benefits provided by the system should be higher than the cost of implementing the system itself. These costs include the expenditures during the development procedure, expenses of training the personnel, and computer and technology costs.

In public works, there are several type of infrastructure to be managed. For each type of infrastructure to be managed, a separate management system is generally needed, such as pavement management systems and bridge management systems. This brings about the problem that, for each individual management system, money should be spent for development and implementation purposes. Most of the time, at the executive level, these infrastructure facilities are operated by different agencies which have separate budgets and their objective is to keep the facility operating at its maximum level of service with the available funds. However, at the administrative level where the total budget is divided among these different infrastructures, it is the decision maker's responsibility to consider all possible effects and make the most cost effective decision.

In addition, even if a particular management system has the most sophisticated models and methodologies to find optimum solutions for a particular infrastructure type, the overall optimum is always affected by other factors. For example utility lines are usually placed under the pavements, and for an effective management of either of the two, this interaction should be considered by the decision maker.

These problems and questions are addressed adequately in the concept of integrated management systems. In this concept, it is assumed that several types of infrastructure can be integrated within a single infrastructure management system which provides decision makers information to examine the impact of various alternative scenarios [Hudson 96]. In addition, the cost of implementing a single system is much less than the total cost of implementing several individual management systems.

The main philosophy behind integrated management systems is to take the common aspects of different management systems and combine them in a single system. The common aspects of management systems can be outlined as follows [Hudson 96];

1. Analysis modules.
2. A central database.
3. Economic analysis models.
4. Optimization models.
5. Graphical interface (e.g. GIS).

Among these aspects, the most important is the database including the physical attribute and condition data for each type of infrastructure. The easiest way of integration is to combine these databases using some common data fields so that information sharing becomes possible between different infrastructures.

The results of the studies conducted by the research team at the University of Texas at Austin show that GIS is a very effective way of integration [Zhang 94]. Figure 2.1 illustrates the concept of an integrated management system.

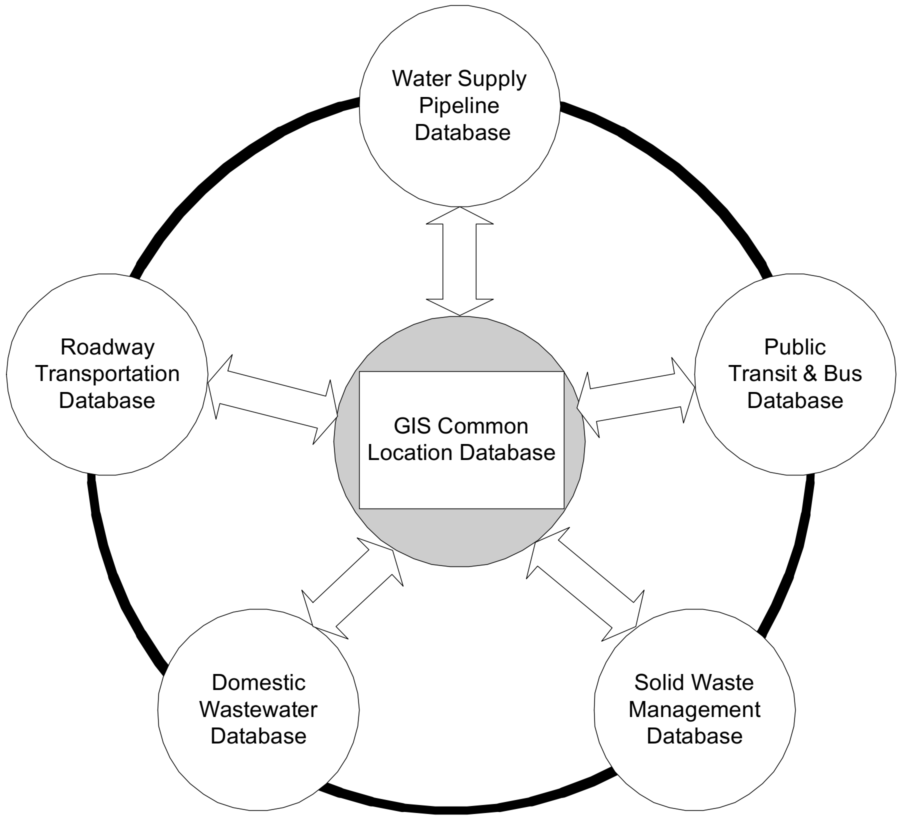


Figure 2.1 Concept of Integrated Management System [Zhang 94]

The concept of the integrated management systems is based on an integrated database and should not be confused with a central combined database [Zhang 94]. In integrated management systems, different databases are linked together by means of some common data fields, for example using geographical data. This is convenient as far as the individual databases are concerned, because the routine data collection procedures and maintaining these databases are not changed in integrated systems. However, in combined databases, all information is stored within a single database which makes it harder to maintain and use.

CHAPTER 3

CURRENT PRACTICES IN PAVEMENT AND BRIDGE MANAGEMENT SYSTEMS

In this chapter, current pavement and bridge management system practices are briefly reviewed and discussed. These systems include: Pontis, Bridgit, TBSS, Visual BMS, URMS, and Visual IMS.

3.1 A NETWORK LEVEL BRIDGE MANAGEMENT SYSTEM: PONTIS

Pontis is defined as a "network level bridge management system which incorporates dynamic, probabilistic models, and a detailed bridge database to predict maintenance and improvement needs, recommend optimal policies, and schedule projects within budget and policy constraints" [Pontis 91]. It is a computerized bridge management tool developed for the Federal Highway Administration.

Pontis uses element-based data, and the important database features include:

1. **Flexibility:** Flexibility of changing procedures and reports as needs change.
2. **Transportability:** It is possible to transfer data between Pontis and any other bridge database.
3. **Capacity:** Can handle inventories of up to 50,000 bridges each having up to 40 elements.
4. **What-if Capacity:** Subjective inputs and adjustments can be entered into the database, and multiple versions of files can be maintained to reflect different scenarios or policy assumptions.
5. **Speed:** High calculation speed.

3.1.1 Structure of Pontis

The six procedures used in Pontis for its analyses are summarized below:

1. **M&R Optimization Procedure of Pontis.**

The objective is to find, for each element in each environment, the policy which minimizes the long-term maintenance funding requirements while keeping the element out of the risk of failure [Pontis 91]. It consists of two interrelated models: The first model determines the optimal action for each condition state of each element, and the second model calculates steady state network conditions if optimal actions are followed. Data required for M&R optimization procedures are:

1) For each element, the set of condition states and for each condition state the set of actions that can be taken.
2) Unit cost of each feasible M&R action.
3) Set of probabilities governing the transition of each element from one condition state to another when an action is taken.

2. **Improvement Optimization Procedure.**

The objective is to maximize the benefits gained in terms of user cost savings. Improvement activities differ from M&R activities in that, as a result of M&R activities, the level of service of the bridge does not change; on the other hand, in improvement activities, the level of service changes.

The benefit cost ratio method is used for this procedure. Bridges are ranked by the ratio of benefits to cost of improvements. The rank list is used to calculate the budget that would be required for improving any set of bridges.

3. **Integration and Program Planning Procedure.**

This procedure provides for scheduling the projects to conform to the budget constraints. It is able to recognize eligibility requirements and funding constraints for specific funding programs, and it can simulate the possibility of future year projects and prioritize them according to their expected B/C ratio. It combines the results of M&R optimization procedure and project programming models and is a powerful for predicting future network conditions, needs, and backlogs. The results of this procedure can be divided into two:

1) Description of M&R and improvement schedules and backlogs, showing how they grow or shrink over time.
2) Detailed first year project list, with the recommended actions.

4. **Condition State and Feasible Action Procedure.**

This procedure gives a statistical profile of the composition and condition of every bridge in the network. All parts of a bridge are characterized by elements made of a particular material. When a bridge is inspected, each element is rated by dividing it into 3 or 5 condition states.

The behavior of each bridge element over time is governed by its environment and random effects of traffic and age. The following environment definitions are used:

1) **Benign:** The condition state of an element is not likely to change due to environmental factors.
2) **Low:** Environmental factors contribute to a decline in the condition of the element at a slow rate.
3) **Moderate:** Environmental factors contribute to a moderate decline in the condition of the element.
4) **Severe:** Environmental factors contribute to the rapid decline in the condition of the element.

5. **Prediction Procedure.**

This procedure estimates deterioration rates for each element and quantifies the uncertainties inherent in such predictions. It consists of two separate models: one quantifies the likelihood that a unit of particular element would make a transition from one condition state to an inferior one; the other model is used to update the prior transition probabilities as data become available.

6. **User Cost Procedure.**

User costs are generated mainly by the level of service deficiencies of a bridge such as narrow width, low vertical clearance, or low load capacity. User costs serve as input to the improvement optimization model which compares the savings in user costs due to replacement or improvement with the cost of investment. User cost is composed of three parts:

1) **Accident costs:** costs due to the accidents under and on the structure.
2) **Vehicle operating costs:** costs related to increased gas or fuel consumption and deterioration of parts resulting from vehicles having to detour due to bridge restrictions.

3) **Travel time costs:** costs of detours for diverted vehicles as a result of weight and height limits posted on the bridge.

3.1.2 Problems in Implementation of Pontis

After Pontis was developed for FHWA, it was tested in departments of transportation (DOTs) in thirteen states. The result of these tests indicated that Pontis is an impressive bridge management system, but it does make significant demands on DOT users [Hearn 93]. It operates with a unique format for coding bridge inspection data which is incompatible with the National Bridge Inventory System (NBIS) database. The reason for this incompatibility is that Pontis employs new elements to describe bridges, defines new condition states for elements, and requires reporting all conditions observed on an element along with the extent of each condition instead of reporting a single average rating value.

In summary, Pontis requires states to change many of their traditional inspection and record keeping procedures. Another problem is that, the Federal government while endorsing Pontis currently, still requires NBI component ratings. Since Pontis uses a different inspection system, either dual inspections will be needed, or an algorithm will have to be used to convert these ratings. Without addressing this issue, an effective implementation of Pontis cannot be carried out [Wells 93].

3.1.3 System Requirements for Pontis

Pontis is a DOS based program developed for use on PCs. It can work on any PC with at least a 80386 processor, 640k RAM, 200 MB hard disk capacity (hard disk capacity requirement may increase or decrease with the number of bridges in the database), and one floppy disk drive. The software is not a user friendly software because it does not have the advantages of Windows-based programs.

3.2 BRIDGIT

Bridgit is a bridge management system software intended to meet the needs of state, local, and other bridge agencies by providing guidance on network level management decisions and project level actions [Bridgit 94]. It was developed under the AASHTO sponsored National Cooperative Highway Research Program (NCHRP Projects 12-28(2)A and 12-28(2)B) and is a microcomputer-based BMS.

Bridgit uses element-based data as does Pontis. There are three different data items in Bridgit:

1. **Segments:** used to subdivide a bridge if it has more than one basic structure type.
2. **Elements:** a segment may contain any number of elements. There are eight categories of elements (substructure, deck, pier, abutments, railing, joint, bearing, culvert). Elements may be in one of the following environmental conditions which affect its decay:
   1) Benign.
   2) Low.
   3) Moderate.
   4) Severe.
3. **Protection Systems:** there are two basic types: overlay and paint.

3.2.1 Implementation of Bridgit

Bridgit was released after Pontis which many DOTs had already begun implementing. After Bridgit was released, several states decided to evaluate it. Currently, Maine DOT is using Bridgit, and Louisiana DOT is using both Pontis and Bridgit. Wisconsin DOT, and the US Army Center for Public Works are also evaluating Bridgit for implementation [Bridgit 94].

3.2.2 System Requirements for Bridgit

Bridgit works on a 80386 compatible computer or better, although large inventories (e.g. 20,000 bridges) may take some time for completion of the optimization procedure on a 386 computer. Bridgit is also a DOS based application like Pontis, but the new version (version 2.0) is a windows compatible software.

3.3 TBSS

TBSS (Texas Bridge Selection System) is a consistent computerized system for the selection of rehabilitation and replacement of bridge projects for Texas [Weissmann 90]. The two objectives of the system can be summarized as follows;

1. To provide a consistent and systematic way of distributing funds statewide.
2. To make project selections for rehabilitation and replacement.

TBSS is a two level closed loop system. The state level applies general selection criteria to the full bridge inventory. The district level takes into account specific local criteria, based on local engineering and planning knowledge about candidate bridge projects, and feeds it back to the state level.

TBSS consists of five different sub-programs as shown in the flowchart depicted in Figure 3.1. The first program is called SURE (Sufficiency Rating Evaluator), which was originally developed and modified for use in TBSS [Boyce 87]. SURE reads appropriate data from BRINSAP database, and applies the FHWA criteria to determine the eligibility of bridges for funding. FHWA criteria require that the Sufficiency Rating should be less than 80 for a bridge to be eligible for Federal Funding [Weissmann 90].

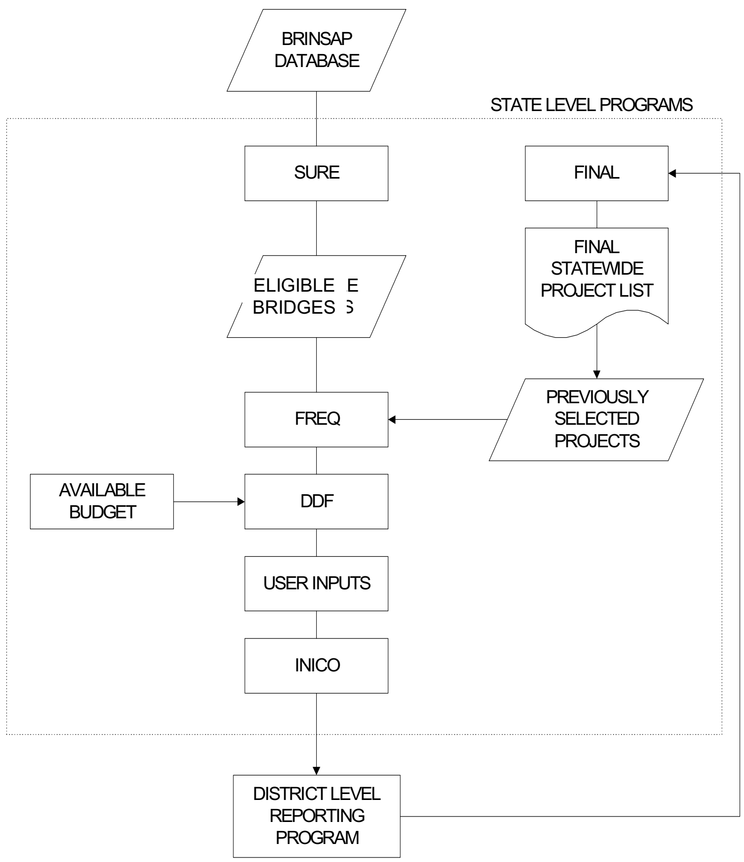


Figure 3.1 The TBSS Flowchart (Source [Weissmann 90])

The next program is called FREQ (Frequencies), and it calculates the frequency distributions of the decision attributes. It processes the federally eligible bridge set sorted by the program SURE. The next program DDF (District Distribution Factors) calculates a budget allocation to districts.

The program INICO (Initially Considered Projects) uses the budget allocations, by district, determined by the DDF program to determine a list of projects to be submitted to the districts for their review. The last program is called FINAL; it combines the results of all programs and prepares the final statewide project list.

3.3.1 Decision Variables Used in TBSS Formulation

Five decision variables are used in the formulation of TBSS [Weissmann 90]:

1. Average Daily Traffic (ADT).
2. Cost per Vehicle (CPV).
3. Minimum Condition (DSS).
4. Sufficiency Rating (SR).
5. Bridge Width Ratio (BWR).

ADT is the average daily traffic over the structure which is a measure of the importance of the bridge relative to the service provided to the users. CPV, cost per vehicle, is defined as the cost of the proposed project divided by the ADT levels. This provides a measure of the cost effectiveness of the project.

The next variable, DSS comprises a minimum of deck, substructure, and superstructure condition ratings. SR is the sufficiency rating index which was created by FHWA [FHWA 87]. It reflects the ability of a structure to remain in service in its present condition. Finally, the last variable is the BWR, which stands for the bridge width ratio. It is defined as the ratio between the existing roadway width and the standardized width.

The formulation of TBSS is based on selecting the projects that maximize the preferences of the decision maker. Concepts of utility theory, which assume that an individual can choose among alternatives available in such a manner that the satisfaction derived from his choice is as large as possible, are applied to achieve this goal.

3.3.2 Implementation of TBSS

TBSS has been developed for and implemented by Texas DOT and is in use. The main advantage of TBSS is that it uses the NBIS data items, which are available in the BRINSAP database. The disadvantage of TBSS is that it does not address the maintenance needs. Currently, it is used as a tool to select the bridges that need to be replaced.

3.4 VISUAL BMS

Visual BMS, developed by Texas Research and Development Incorporated (TRDI), is a software capable of sorting, retrieving, and processing bridge related condition and inventory data in a roadway network [TRDI 96]. It allows the user to analyze the current condition and expected needs for the bridge population.

The Visual BMS software is a Microsoft Windows compatible program with graphical user interfaces common to all Windows applications. It is designed as a client server application to work ideally within a coordinated suite of transportation management systems. The software can handle a wide range of predefined and user defined reporting and graphical presentation capabilities, displaying results that are clearly understood by the user.

3.4.1 Visual BMS Overview

Visual BMS provides many capabilities in network level bridge management. The network optimization techniques use Markov transition probabilities to estimate bridge network performance and perform true optimization to determine the distribution of the budget between the different treatment strategies, subject to constraints of desired bridge levels of service or future budget levels.

In addition to determining the overall budget consequences, the bridge management system assists with the selection of the location, recommended treatment, and timing of projects. Visual BMS offers two approaches for model formulation: multi-year prioritization (Bridgit methodology) and optimization (Pontis methodology).

Visual BMS provides the following analysis capabilities:

1. Condition and load capacity analysis.
2. Level of service analysis.
3. Performance and deterioration model development.
4. Remaining service life analysis.
5. Network optimization.
6. Multi-year incremental benefit cost prioritization.
7. Data summarization and statistical analysis.
8. Extensive report and graphics presentation development.

Although Visual BMS is still under development, some states in the US have evaluated it for their bridge management systems. The main advantage of Visual BMS is that the software is Windows compliant, very easy to implement and user friendly. On the other hand, like Pontis and Bridgit, Visual BMS also uses element-based bridge definition which is not compatible with the NBIS database.

3.5 URBAN ROADWAY MANAGEMENT SYSTEM (URMS)

URMS is a computer program which assists decision makers in managing pavements at both the network level and the project level [Chen 93]. It consists of four subsystems:

1. **Planning:** To identify and select the most cost-effective maintenance and rehabilitation (M&R) projects in a street network, and help develop annual M&R programs.
2. **Design:** To select pavement materials and determine layer thicknesses for those sections selected for overlay or reconstruction.
3. **Construction:** To schedule overlay and reconstruction projects.
4. **Maintenance:** To select effective distress repair methods using a simplified expert system.

URMS is a simple program with a size of approximately 350 kb. It has only one executable which contains all the subsystems. It is a flexible software from the user's point of view, because all distress types, M&R strategies and model parameters can be modified by the user while running the software. Though it was developed using the graphical user interface (GUI) technology which makes it user friendly under the DOS environment, but it is not a Windows-based program, works under DOS environment.

3.5.1 URMS Subsystems

Figure 3.2 shows the overall data flow diagram of URMS. First, an M&R strategy is assigned to each section, and the total cost for all M&R sections in the street network is calculated. If the total cost is greater than the budget, prioritization is performed and the M&R candidate projects are listed. The M&R strategies consist of four types: reconstruction, overlay, routine maintenance and do nothing.

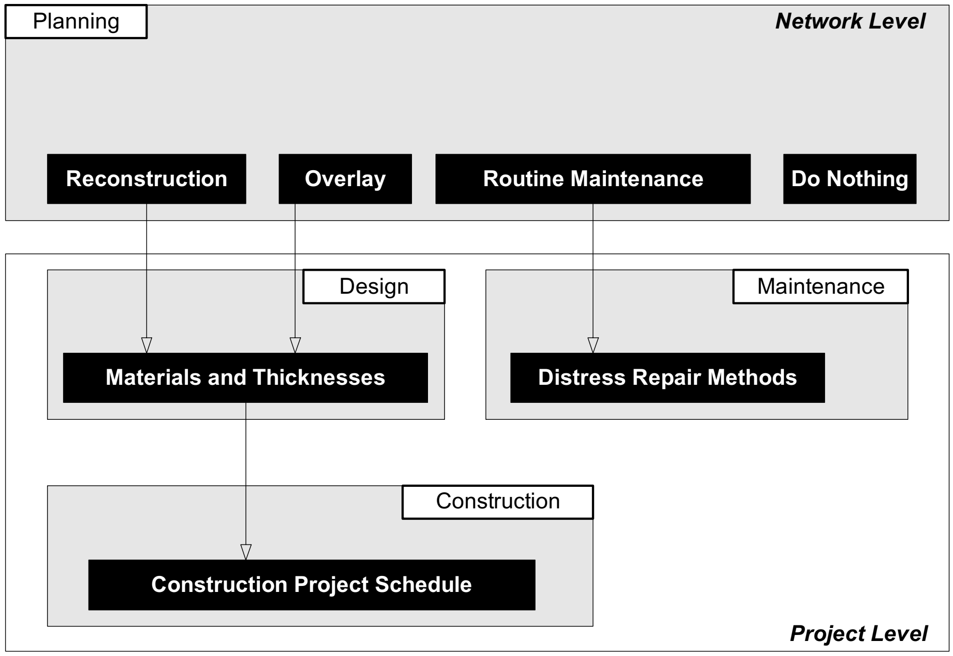


Figure 3.2 Overall Data Flow Diagram of URMS

If the M&R result is reconstruction or overlay, URMS calculates the new pavement thicknesses using its design subsystem. There are two options in the design subsystem: an optimal design to get the least cost solutions, and the conventional design which allows the user to design layer thicknesses interactively.

The construction subsystem helps schedule the constructions for overlay and reconstruction activities. URMS uses the CPM/PERT (Critical Path Method, Program Evaluation and Review Technique) for this purpose. URMS generates optimal alternatives for either minimizing total project cost or minimizing expedited cost for a given expedited time.

The maintenance subsystem contains a knowledge base that is used in consulting for selecting distress repair methods for different types of distress. The knowledge base is first formed by the user by entering distress types and possible distress repair methods. After the knowledge base is established, the user can get advice from the program on possible maintenance strategies for a specific type of distress.

3.5.2 Implementation of URMS

After URMS was developed, a further study was conducted to implement URMS in small to medium sized cities [Sohail 96]. The scope of the implementation was intended to be used nationwide; however, because of limited funds and time, the implementation was restricted to a few Texas cities. The implementation process was carried out at two levels. At the first level, the city of Lampasas was directly assisted and the project staff worked closely with the city on implementation. At the second level, all other cities which showed considerable interest in implementation were offered assistance via telephone. Due to funding constraints, the complete second level implementation was possible only for the city of Terrell.

As a result of the implementation study, URMS was implemented in the two cities without any extensive extra data collection effort by city personnel. The implementation was successful in both cities and demonstrated that URMS is simple enough to be adopted by small cities with minimum support.

3.6 VISUAL IMS

Visual IMS, developed at the University of Texas at Austin, is a GIS based and multimedia integrated infrastructure management system [Zhang 96]. It was developed primarily for application by public work agencies, and includes three subsystems: roads, water supply, and wastewater.

Visual IMS can handle operations of both numerical data and graphical objects. It is a Windows application and is facilitated with a graphical user interface (GUI). It is programmed in Avenue, an object oriented GIS programming language. It works in IBM compatible computers with a 80386 or higher processor and at least 20 MB of hard disk space available on the computer.

3.6.1 Operation of Visual IMS

The operation of Visual IMS is divided into two parts:

1. Analytical operations based on decision models.
2. Visualizing and mapping operations based on GIS and its integration with multimedia.

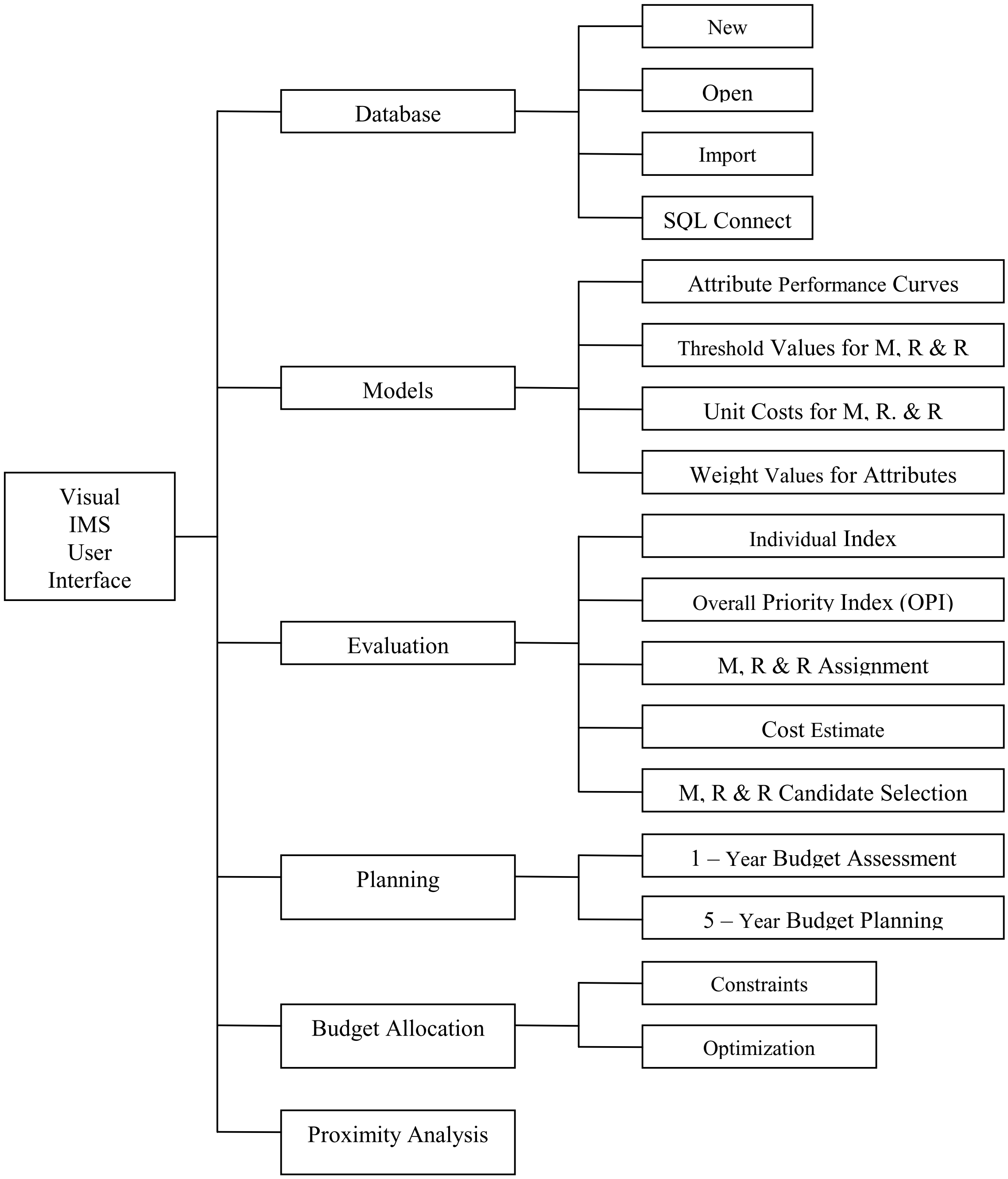


Figure 3.3 Analytical Functions in Visual IMS (Source [Zhang 96])

As shown in Figure 3.3, the analytical operations of Visual IMS consist of six different modules. The database module provides editing, entering or importing the data. Five groups of input data are required by Visual IMS: 1) geographic data that defines the coordinates of particular infrastructure features, 2) inventory data that provides an accounting of the physical infrastructure attributes, 3) condition data to be used in evaluation procedures, 4) cost data, and 5) multimedia data.

Visual IMS is a generic system rather than an application specific system. The user can adjust the models used in the procedures to incorporate local experience and practice via the model setup feature. It has a user friendly model editor which allow users to modify model curves interactively on the computer screen.

The evaluation module of Visual IMS provides a method for selecting and ranking the candidate projects for M & R. Visual IMS uses a weighted overall priority index (OPI) model in selecting M & R projects. The cost of each project is calculated and the total cost is compared with the available budget. If the total cost exceeds the budget, the selected projects are prioritized to form the final list.

The planning module provides a multi-year analysis and is based on the performance index (PI) prediction model [Zhang 96]. The budget allocation module provides an optimum budget allocation among three different infrastructure subsystems, namely road, water supply, and wastewater.

One of the most important analysis functions in the Visual IMS is the proximity analysis. This function provides a tool for the user to schedule all M & R activities for the related infrastructures in a systematic and coordinated manner, and thus will increase the overall efficiency of M & R. As one of the most important features of integrated management systems, data sharing in Visual IMS is based on geographic location.

Visualizing and mapping operations of Visual IMS provides six main functions [Zhang 96]:

1. Map display.
2. Feature Selection.
3. Spatial query.
4. Spatial analysis.
5. Multimedia link.
6. Map production.

3.6.2 Implementation of Visual IMS

In order to demonstrate how an IMS can be used as an enhancement to existing management systems, Visual IMS was implemented in several case studies [Hudson 96]. These studies include: 1) Montana PMS in which Visual IMS is used as a supporting tool to the existing PMS, and 2) TMS for Cobb County DOT in which the concept of Visual IMS was applied to develop a total management system that includes six management modules: road inventory, traffic congestion, bridge maintenance, pavement management, traffic signal, and work order scheduling.

The results of the case studies show that integrated management systems provide enhancements in existing management systems, and GIS is the proper integration platform for managing infrastructure facilities.

CHAPTER 4

MODULES AND FORMULATION

Formulation of the modules in the new system can be summarized in three categories:

1. Formulation of the pavement management module.
2. Formulation of the bridge management module.
3. Formulation of the integrated system.

4.1 FORMULATION OF THE PAVEMENT MANAGEMENT MODULE

In the information processing phase of the pavement management systems, the important task is the selection of proper maintenance and rehabilitation (M&R) strategy which is economically feasible. There are several approaches in selecting M&R alternatives and these can be summarized as follows [Shahin 94]:

1. **Ad Hoc Approach:** In this approach, the user applies the M&R alternative that provides the best solution based only on past practice. This approach may limit the number of M&R alternatives to the user's knowledge and experience, and may not select the most economic alternative.
2. **Present Condition Approach:** In this approach, the pavement is evaluated using various condition indicators which usually include distress, deflection, roughness and skid resistance, and an M&R alternative is selected to improve the condition. The possible alternatives are not compared in terms of life-cycle costs in this approach. A major advantage of this approach is that the selected M&R alternative corrects the deficiencies found in the pavement.
3. **Life-Cycle Approach:** This approach ensures selection of the most economic M&R strategy on a life-cycle cost basis which requires both an in-depth evaluation of the pavement under consideration and prediction of its future condition. In order for this approach to produce good results, future maintenance activities and their costs should be accurately determined.

In this study, the present condition approach is used to select the M&R alternative for pavements. The procedure is a one year analysis procedure which addresses solutions for the current deficiencies. It is important to realize that another important task in pavement management is the distribution of a limited amount of funds for performing these M&R actions.

Most of the time, the available budget is much lower than what is required to correct all the deficiencies. This brings about the need for a ranking or prioritization procedure to distribute the available funds among possible M&R candidates.

The number and type of pavement M&R strategies in pavement management vary from agency to agency. The most popular M&R strategies include: do nothing, routine maintenance (repair distresses), thin overlay, thick overlay, and reconstruction [Chen 93]. In this study, the decision tree method is used to assign the M&R strategy. The decision tree method is a very popular method where the variables used in the decision making process are taken into account step by step. Figure 4.1 uses three decision variables, and the tree structure selects the thick overlay strategy.

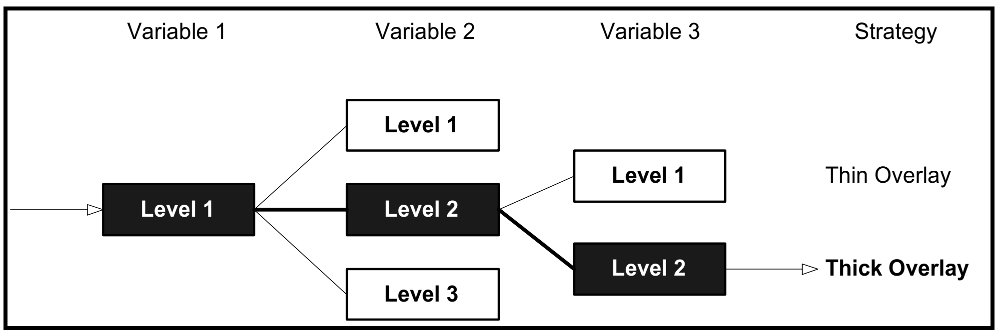


Figure 4.1 Decision Tree Method (Source [Chen 93])

Three decision variables, pavement condition index (PCI), pavement age (AGE) and average daily traffic (ADT), are used in M&R strategy selection for pavements. Each variable is divided into five levels as shown in Table 4.1.

Table 4.1 Classification of Decision Variables

Levels	PCI	AGE	ADT
1	Bad	Very Old	Very Heavy
2	Poor	Old	Heavy
3	Fair	Average	Medium
4	Good	New	Light
5	Excellent	Very New	Very Light

Using these three decision variables with five levels for each result yields a total of 125 (5x5x5=125) M&R alternatives. Such a large number of alternatives is not very practical in the implementation process [Chen 93]. The total number of M&R strategies is therefore reduced to a reasonable level. Because most cities use 10 to 20 M&R strategies for budgeting and programming, the first and last two levels of variables are combined together in one level as shown in Figure 4.2.

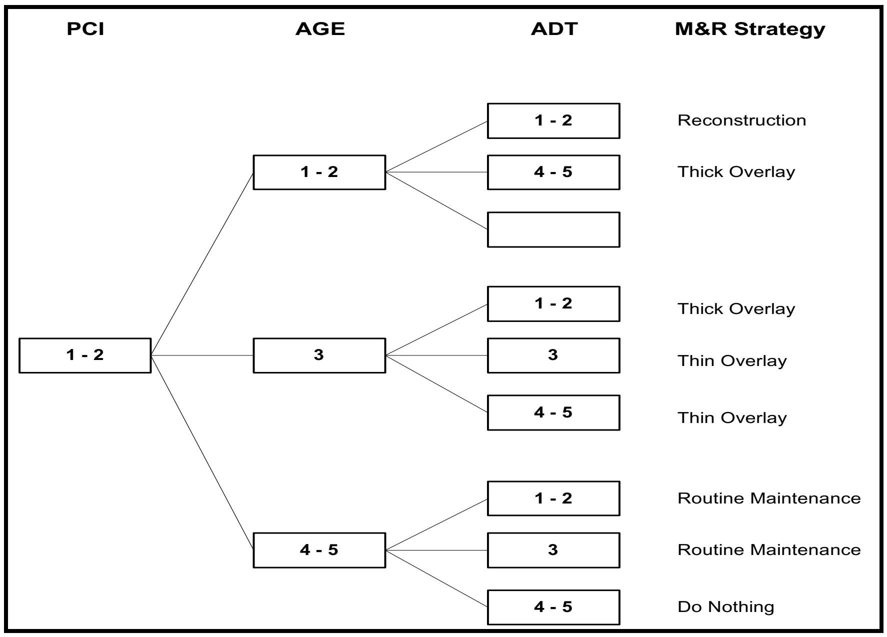


Figure 4.2 One Branch of the Decision Tree (Source [Chen 93])

From Figure 4.2, Level 1 ("Bad") and Level 2 ("Poor") of PCI are combined to represent a single PCI level. This is not very far from reality, because these levels are usually determined by subjective methods, mainly based on the engineer's experience. Similarly, Level 4 ("New") and Level 5 ("Very New") of AGE are combined into one level.

The overall pavement management formulation can be explained with the diagram shown in Figure 4.3. First, certain statistic values are calculated for the pavement network. These values are then used in selecting the levels of decision variables (PCI, AGE, ADT) by the user. The decision tree is then applied to the management sections sorted by a user defined criteria, and the M&R strategies are assigned to each section with an estimated cost. The procedure continues until the accumulated cost is greater than the available budget.

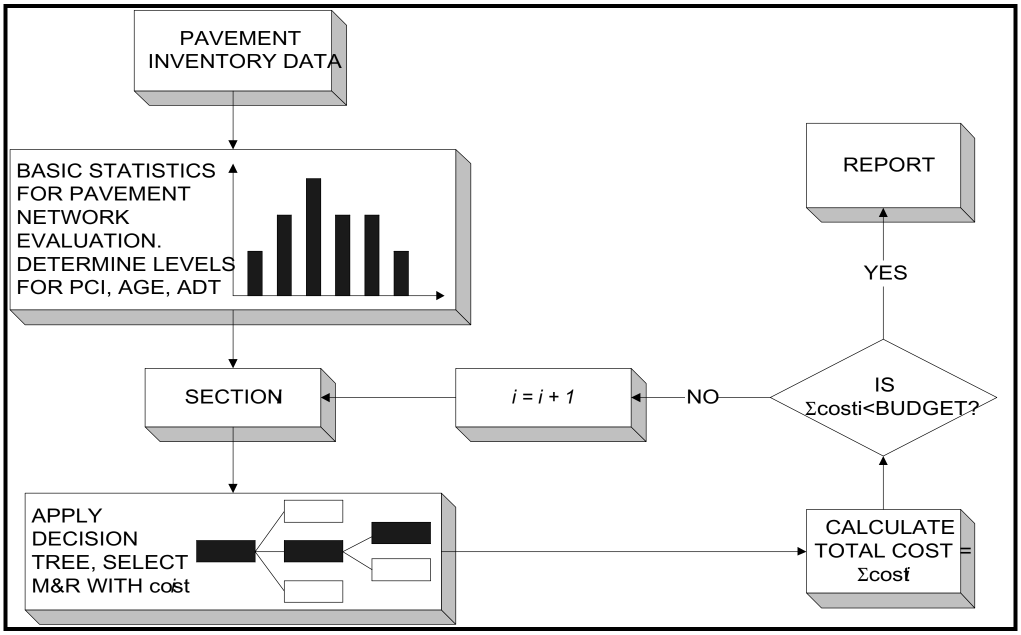


Figure 4.3 Pavement Management Formulation

4.2 FORMULATION OF THE BRIDGE MANAGEMENT MODULE

Bridges are more complicated facilities as compared to pavements. There are many elements in bridges, such as piers, curbs, girders, and deck. Most of these elements exist in different environments. For example, in water crossing bridges, the piers are exposed to water, however the bridge deck is exposed to air. As far as the structural functions are concerned, piers and deck carry the traffic loads, whereas curbs are for drainage purposes.

A good bridge management system should be capable of answering the questions related to all of these elements. For example, it is always good for the decision maker to know the condition of each element and the total quantity of elements in each condition level. Although current bridge management tools like Pontis and Bridget can perform these functions, the need for element-based condition data is delaying the successful implementation of these systems [Pontis 91, Bridgit 94].

The National Bridge Inventorying System (NBIS) database includes physical attribute data for all of the elements in a bridge, but it does not include comprehensive condition data. Condition data are available only for the main bridge elements, namely, deck, substructure, superstructure, roadway approach, channel protections, and culverts [NBIS 96]. In this study, in order to fully use the NBIS database, the deck, substructure, and superstructure are used as the main elements for bridges, and M&R project assignment is made according to the conditions of these three elements.

Another dimension in bridge management is the operational restrictions and bridge inspections. Bridges with inadequate load carrying capacity, for example, are posted for some level of traffic loads rather than trying to improve the bridge capacity. Bridge inspection is an important issue because the consequence of bridge failures is more dangerous than that of pavement failures. Some features in bridges such as “Scour” are key factors in bridge structure stability, and should be inspected with care.

For these reasons, the bridge management system consists of three parts in this study:

1. A flagging mechanism for identifying the bridges with certain deficiencies in order to schedule special inspections, or assign operational restrictions.
2. A decision tree for assigning specific M&R projects for the three main bridge elements: deck, substructure, and superstructure.
3. A decision tree for assigning operational restrictions (like bridge posting) and schedule special inspections.

The formulation takes full advantage of the NBIS data items that are collected every two years nationwide. Since many cities do not maintain any inventory for their bridges (cities take the inventory data from the NBIS data which is collected by DOT’s), the formulation does not include a detailed inventorying module. It is important to note that several safety data items are emphasized through the flagging mechanism and some basic bridge identification data are included for proper reporting.

4.2.1 Flagging Mechanism

The flagging mechanism provides a filter to pop out bridges with certain deficiencies. It is good for the decision makers to have access to a short list of deficient bridges, for example scour critical bridges, so that special inspection for these bridges can be scheduled. In addition, this mechanism can also be used as a network level selection tool to reduce the total number of bridges to be evaluated.

The variables used in decision making for operational restrictions and special inspection scheduling are:

1. **Traffic Safety Features:** These features include bridge railings, transitions, approach guardrails, and approach guardrail ends. The NBIS database contains information about either a traffic safety feature meets accepted standards or not. Bridges that do not meet the standards should be selected.
2. **Scour Critical Bridges:** In the NBIS database there is a one digit scour critical bridge code varying from 0 to 9, 0 meaning that the bridge is scour critical. Bridges with scour value less than a user-defined value should be selected for special inspection (in the NBIS database, if this value is 3, the bridge is defined to be scour critical).
3. **Bridge Width:** This factor is important if the bridge width is less than the roadway width, or the number of lanes on a bridge is less than that of the roadway. If the bridge width is less than a user-defined value, the bridge can be selected for special inspection scheduling.
4. **Loading Restriction:** The NBIS database contains a loading restriction data item varying from 3 to 9. Bridges with any type of loading restriction should be selected (for bridge posting).

The following variables are used for a network level mechanism in addition to the above variables:

1. **Historical Significance:** The NBIS database contains information about the historical significance of a bridge. For example a bridge may be a unique example of the history of engineering, or it might be associated with a historical property or area. There are five levels for historical significance in the NBIS database (1 being the most significant, 5 being the least). Bridges with historical significance levels less then a user-defined level should be selected as eligible bridges for maintenance and rehabilitation funds.
2. **Bridge Length:** The longer the bridge, the more important it is. Critical length (L_{cr}) that separates long bridges from short ones is to be determined by the user. For example, in a bridge network where the longest bridge is 1000 ft, this number may be 500 or 600 ft. However, in a bridge network where the longest bridge is 3000 ft, this number may be 1000 or 1500 ft.
3. **ADT:** Another factor that shows the importance of a bridge is the Average Daily Traffic (ADT) value. The more traffic the bridge serves, the more important it is. The

critical ADT (ADT_{cr}) value for separating low and high ADT will be determined by the user.

4. **Condition:** The worse the condition of a bridge, the more important it is. One of the main objectives of a bridge management system is to improve the bridge conditions to acceptable levels. In the NBIS database there are condition data for deck, substructure and superstructure. To take into account the worst situation, the minimum of these conditions is used as the representative condition value for the whole bridge. The critical condition value (C_{cr}) should be determined by the user.

The bridge management formulation is a simple filtering mechanism, such that the bridges satisfying these criteria will be selected as candidates for M&R project assignment, operational restrictions, or special inspection scheduling. In most of the bridge management practices today, the limits set for these variables are determined by the managers (i.e. the manager will decide which values of ADT, length, or condition will form the selection criteria). The managers (in most cases. the engineers) usually use their personal experience and judgment, or use some generally accepted numbers. Most of the time, the values set by these engineers are rounded off, for example most engineers choose the ADT values of 1000 and 1500 as the limiting values between low, medium and high traffic. This problem can be overcome from the distributions of these variables and by calculating some simple statistic values. Rounded off

Table 4.2 lists the variables used in flagging filtering (or network level selection) mechanism. Each variable has two levels: either critical or non-critical.

Table 4.2 Classification of Decision Variables for Flagging Filter

	Level 1	**Level 2**
Historical Significance	Significant	Non-Significant
Bridge Length	Long	Short
ADT	Heavy	Light
Condition	Poor	Good
Safety Features	O.K.	Not O.K.
Scour Critical	Critical	Non-Critical
Deck Width	Narrow	Wide
Load Restriction	Restricted	Non-Restricted

According to Table 4.2, for example, the user can generate a list of scour critical bridges, bridges with load restrictions, narrow bridges, etc. This list can also be used in further analyses.

The flagging filtering mechanism is illustrated with the diagram shown in Figure 4.4. First, the basic statistics are calculated using the bridge inventory data. These values are then used to select the critical values for the variables used in the filtering mechanism. Next, each bridge within the network is filtered using this criteria. The result of the filter is used either for reporting purposes or as a network level filter prior to the application of decision trees to select specific M&R projects or assigning special inspections.

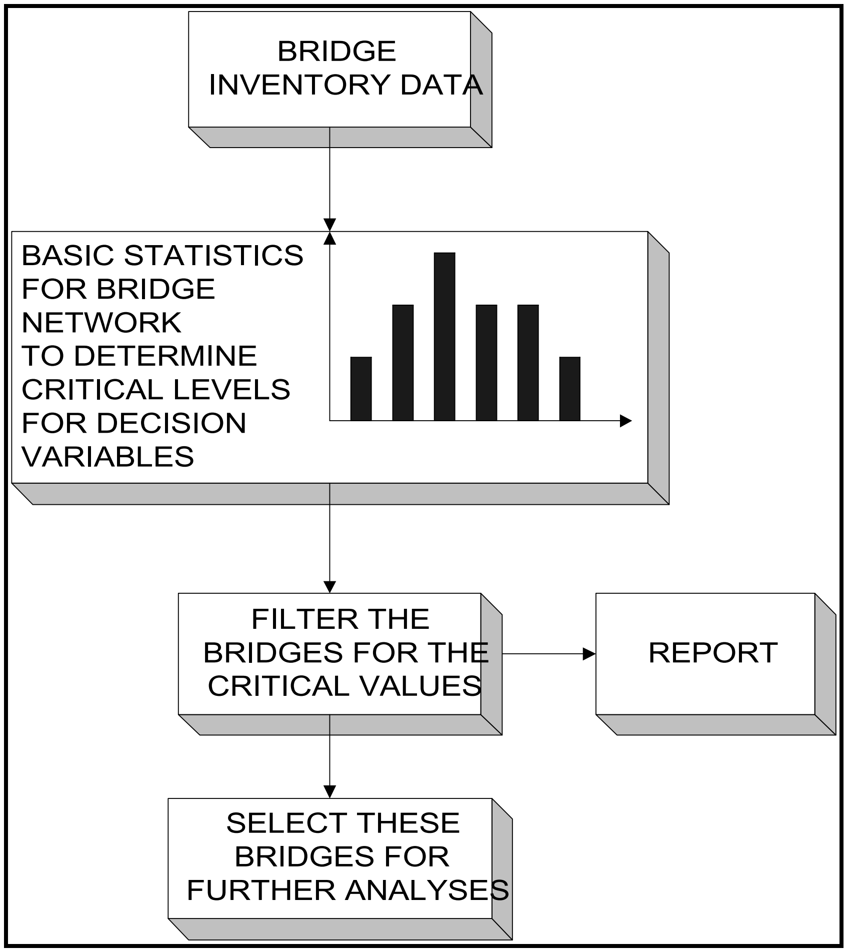


Figure 4.4 Flagging Filtering Mechanism

4.2.2 Decision Tree for Assigning Operational Restrictions and Special Inspections

Bridge management formulation is also based on the present condition approach as in the pavement management formulation. The solution is reached by means of decision trees. The first decision tree is for assigning operational restrictions and scheduling special inspections. Four variables are used in this decision tree as shown in Table 4.3.

Table 4.3 Classification of Decision Variables for First Decision Tree

Levels	Scour Code	Deck Width	Load Restriction	Safety Feature
1	Very Critical	Large	Restricted	Not O.K.
2	Critical	Narrow	Non-restricted	O.K.
3	Non-critical			

The decision tree structure for assigning operational restrictions and scheduling special inspections is shown in Figure 4.5. It should be noted that the activities shown in this figure are examples of possible activities. In a real system, the user has the flexibility to set any kind of activity that is applicable to a particular agency.

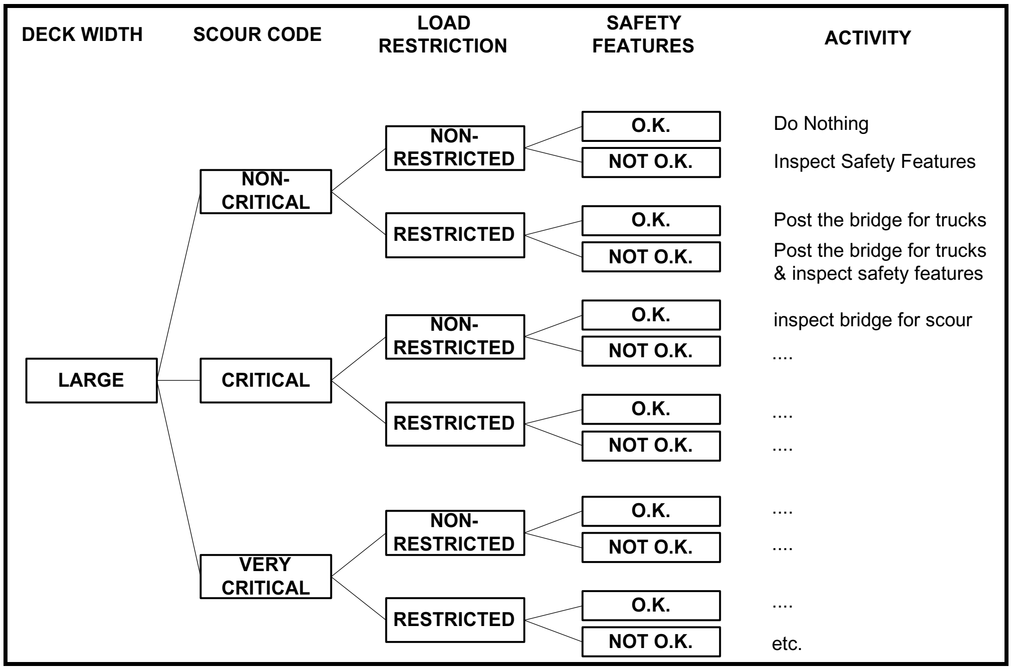


Figure 4.5 One Branch of the Decision Tree for Operational Restrictions and Special Inspections

4.2.3 Decision Tree for Assigning Specific M&R Projects

As stated earlier, three main elements of bridges are used in a specific M&R project assignment: deck, substructure and superstructure. The M&R decision tree uses the condition values of these three elements as the decision variables. There are three levels for each variable as shown in Table 4.4.

Table 4.4 Classification of Decision Variables for M&R Decision Tree

Levels	Deck Condition	Substructure Condition	Superstructure Condition
1	Bad	Bad	Bad
2	Marginal	Marginal	Marginal
3	Good	Good	Good

The decision tree structure for selecting specific M&R projects is shown in Figure 4.6. The user can change the M&R strategies for each branch of the tree.

Possible M&R activities for bridges can be defined as follows;

1. **Repair the Deck:** All minor maintenance activities for deck, including patching of the deck surface and/or sidewalks, general cleaning of the deck, cleaning of expansion joints, etc.
2. **Reconstruct the Deck:** This action should be taken if the deck and superstructure can be treated separately (i.e. reconstruction of the deck does not require reconstruction or replacement of the superstructure)
3. **Repair the Superstructure:** All minor maintenance activities for the superstructure, including painting the guardrails, cleaning the bearings and the gutters, etc.

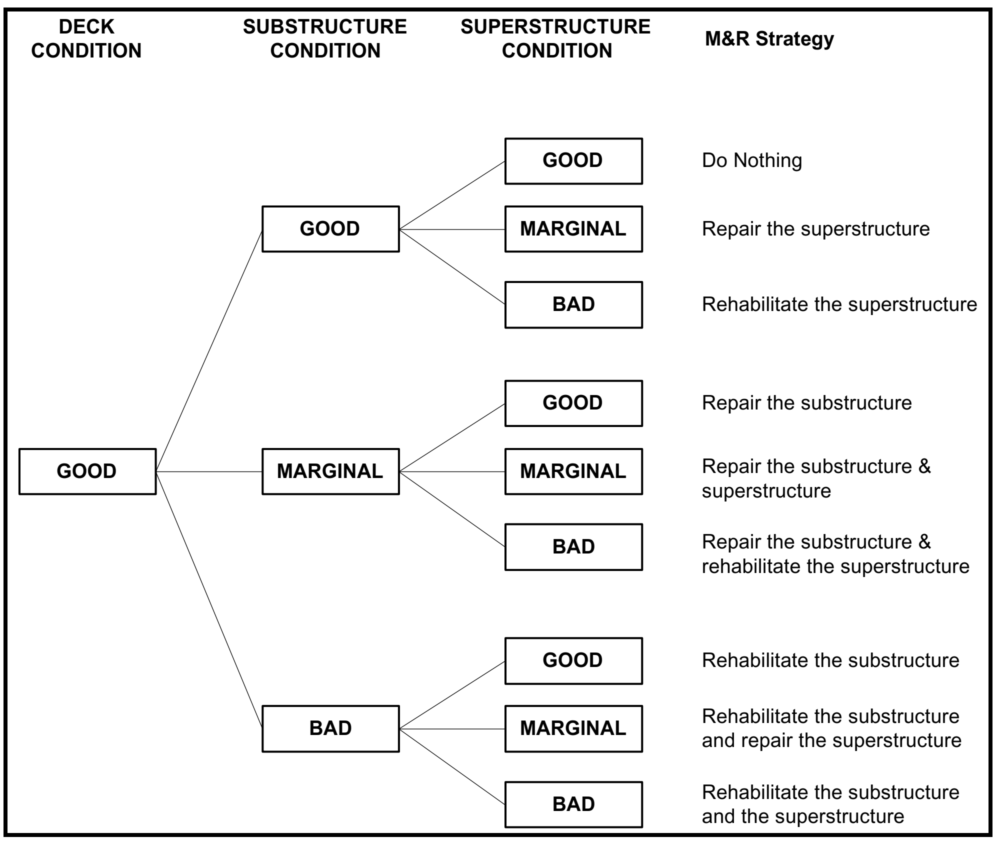


Figure 4.6 One Branch of the Decision Tree for M&R Project Selection

4. **Rehabilitate the Superstructure:** Activities that improve the condition of the superstructure, including replacement of guardrails, replacing the bearings, reconstructing the gutters, etc.
5. **Repair the Substructure:** All minor maintenance activities for the substructure, including minor repairs for piers, abutments, foundations, etc.
6. **Rehabilitate the Substructure:** Activities that improve the condition of the substructure, including the rehabilitation of foundations, piers, and abutments.
7. **Repair the Bridge:** Repair of all the elements as described above.
8. **Rehabilitate the Bridge:** Rehabilitation of all the elements as described above.
9. **Replace the Bridge:** Replacing the bridge with a new one.

The overall bridge management formulation used in this study can be explained with the diagram shown in Figure 4.7.

4.3 FORMULATION OF THE INTEGRATED SYSTEM

As stated earlier, one of the important issues in integrated management systems is the development of common evaluation indices or criteria in order to be able to compare different types of infrastructure. Most of the time, at the administrative level, it is necessary to split the total budget among different types of infrastructure for maintenance and rehabilitation purposes. For example, it is important for an agency's manager to properly divide the total funds between the pavement and bridge departments. In order to be able to make such a budget allocation, decision makers should have tools to compare different infrastructure on a common scale.

In this study, in order to evaluate bridges and pavements simultaneously, a common priority index is developed. In developing this index, the evaluation criteria used in pavement and bridge management systems are examined to see if they can be combined. The objectives of a bridge management system can be generalized to fit all infrastructure and can be used in developing a combined index for bridges and pavements [Weissmann 90]. Figure 4.8 depicts a diagram that shows the objectives of an infrastructure management system and associated attributes to achieve these objectives.

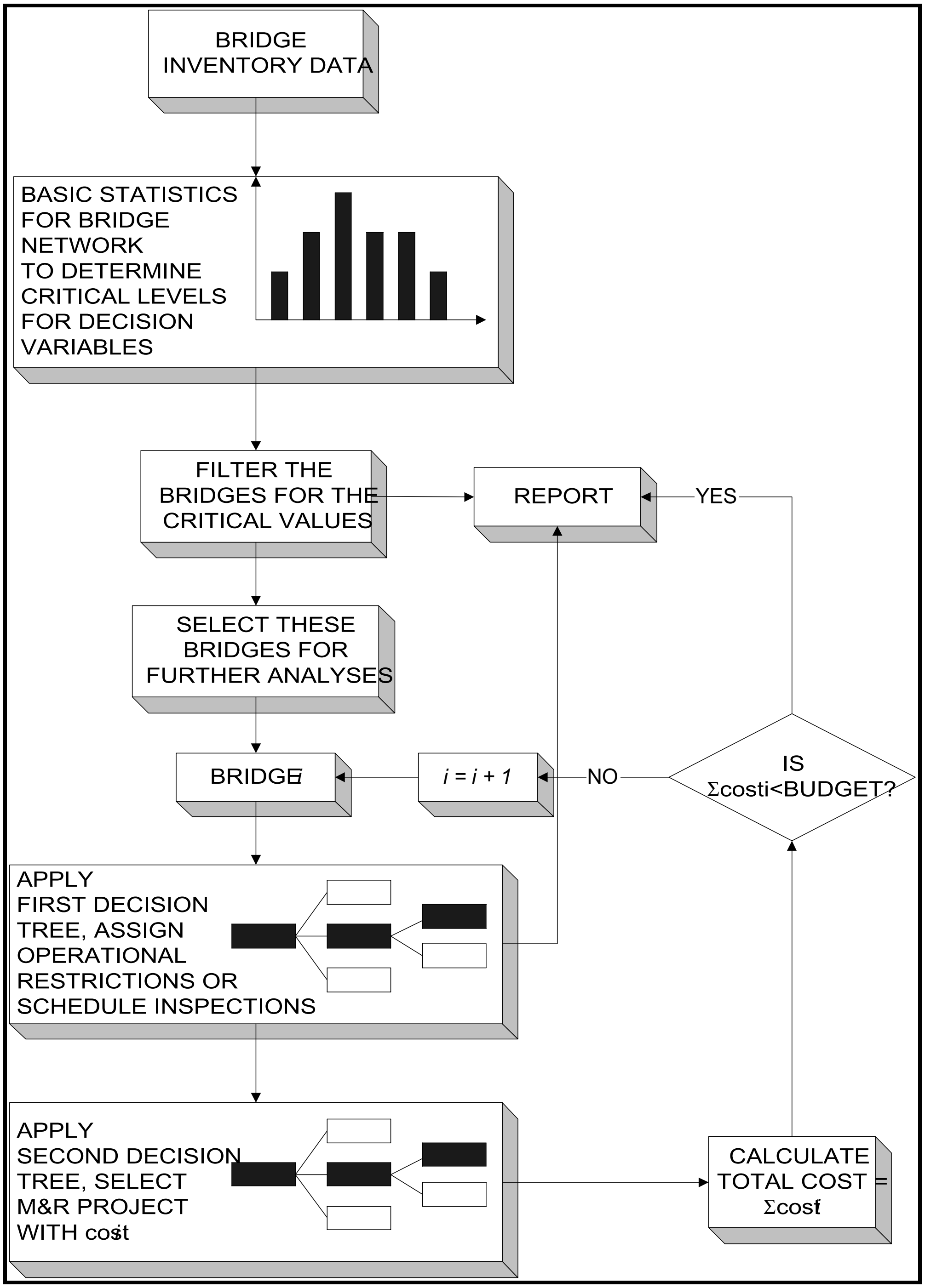


Figure 4.7 Bridge Management Formulation

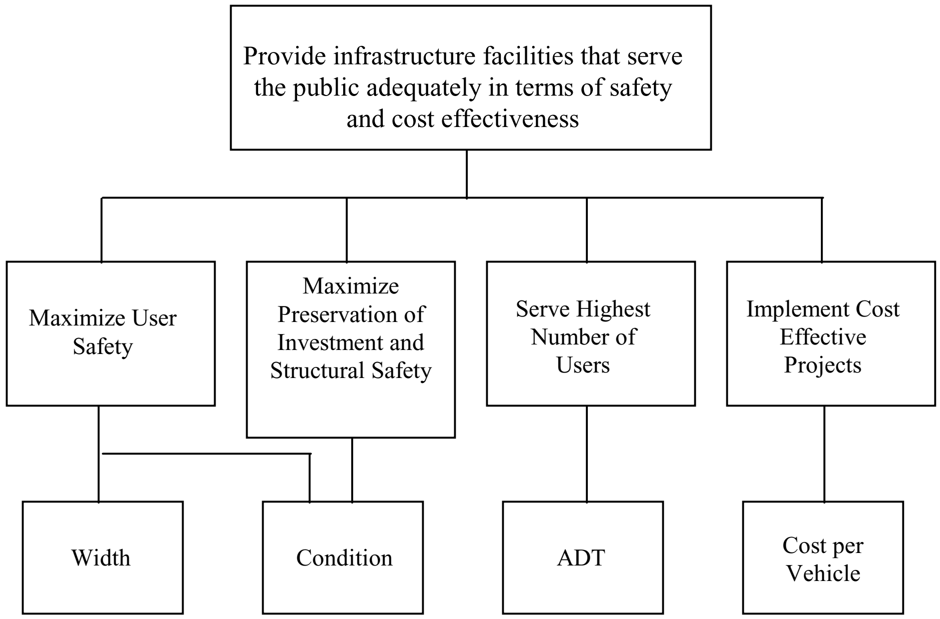


Figure 4.8 Objectives and Associated Attributes [Weissmann 90]

In this diagram, the major attributes for achieving the objectives are depicted on the last line, including the width, condition, ADT and cost per vehicle. Although Weissmann's study is originally for bridges, Chen's study shows that these attributes are not much different for pavements. Age, ADT and condition index attributes are used to evaluate pavements in this study [Chen 93].

With some modifications in the bridge and pavement attributes, a combined evaluation index is developed for both bridges and pavements. In the combined index, ADT, age, width and condition attributes are used. There is a need to modify the condition attributes of bridges so that they can be compatible with the pavement condition index. The condition of a bridge is represented by the minimum conditions of the deck, substructure and superstructure. Bridge conditions are evaluated with numbers between 0 and 9, whereas the pavement condition index is evaluated between 0 and 100. To evaluate the two attributes on a common scale, bridge condition values are multiplied by 100/9 (or 11.1).

Age is the difference between the current year and the year of construction for a bridge; or the difference between the current year and the year of last major rehabilitation action or construction for a pavement section. ADT is the daily mixed traffic value, and width is the width of bridge deck or pavement section. Each attribute is represented in three levels.

Two matrices are used in the calculation of the priority index as shown in Figure 4.9. The priority index is the summation of the two corresponding values coming from each matrix.

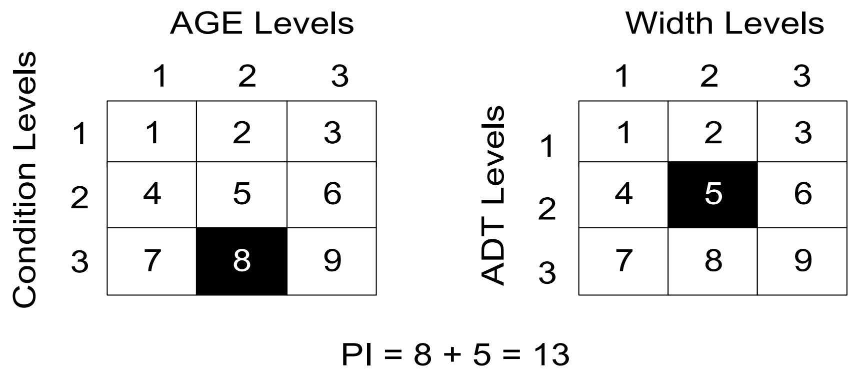


Figure 4.9 Priority Index Matrices

The levels for variables are defined as shown in Table 4.5.

Table 4.5 Levels of Variables for Prioritization Procedure

Levels	Condition	Age	ADT	Width
1	Bad	Old	Heavy	Narrow
2	Fair	Medium	Medium	Medium
3	Good	New	Light	Wide

Figure 4.9 also shows an example for the calculation of priority index (PI); where the condition level of 3, age level of 2, ADT level of 2 and width level of 2, the priority index is 13. Small PI means that the structure should be given higher priority when considering the M&R project assignment.

CHAPTER 5
SYSTEM DESIGN

The key factor in an effective infrastructure management system is a relational, centralized database. All the modules and functions within the system communicate with this database to perform the analyses by means of a computerized system. Therefore there are two important parts in system design:

1. Database design, which includes the design of database tables and their relations.
2. Overall system design, which includes the data flow diagrams through the different models of the system.

5.1 DATABASE DESIGN

Database design includes defining the data-items that should be included in the database tables which are used to store and manipulate the data. There are two types of database tables in the system:

1. Raw data tables.
2. Calculated data tables.

5.1.1 Raw Data Tables

Raw data tables include the inventory and condition data related to the infrastructures, as well as any user inputted data. This is the type of data that is collected in the field to be put into the system as the initial input. One of the major functions of an infrastructure management system is the storage and maintenance of the inventory data [Hudson 87]. Raw data tables provide functions like entering, modifying or viewing the inventory data. They also provide input for the analysis modules of infrastructure management systems.

The level of data detail should be well defined when setting up an infrastructure management system. If the purpose is to answer all possible questions using the management system, then the data should be as detailed as possible. However, the problem of collecting, inventorying and managing these detailed data may become very difficult and the system may be unfeasible for implementation due to the excessive data requirements. When designing an infrastructure management system, efforts should be given to the use of the existing data. A bridge management system tool like Pontis [Pontis 91] requires so many more data items than what is available on the National Bridge Inventorying System (NBIS) that it is almost economically unfeasible for many agencies to implement it.

Since this study includes two types of infrastructure, there are two sets of database tables: one for pavements and the other for bridges. The data items used are those which are currently available. For example, the NBIS database, which is available for each transportation department and updated every two years, is used as the bridge data source [NBIS 96]. Tables A1 through A15 in Appendix A show the raw data tables and data item definitions for these tables. The "Null" column indicates whether this item can be null or not, such that if this column is "N", there should certainly be a value for each record on this column. The "Key" column indicates the ID for the data table.

With the exception of the inventory or condition data for the bridge or pavement infrastructure, there are two other raw data tables. One is the user information table, which contains information related to the analyzer of the system, as shown in Table A16 of Appendix A. The information stored in this table is used for reporting purposes. The second table is used for storing the budget information, as shown in Table A17 of Appendix A.

5.1.2 Calculated Data Tables

The calculated data tables contain the results from all calculations in the system. The working principle of the system is straight forward: the setup information, raw inventory, and condition data are combined together with the analyses formulations to complete the analyses. During these procedures data tables are needed to store the combined and calculated data. Tables A18 through A21, in Appendix A, summarize these tables and data item definitions for them. Although these tables are called "Calculated Data Tables", they do not necessarily have to include all the results of calculations. Combinations of several raw data tables are also called calculated tables although there is no actual calculation involved.

5.2 Overall System Design

System design defines the flow of information through the database tables. Generally, the physical attribute and condition data must be provided by the user through inventory and condition tables. The decision tree structures, critical values for variables to be used in decision trees, and M&R actions and their unit costs are defined in the setup tables. Proper setup information is combined together with the physical attribute and condition data to perform the analyses. The overall system structure is depicted with the diagram in Figure 5.1.

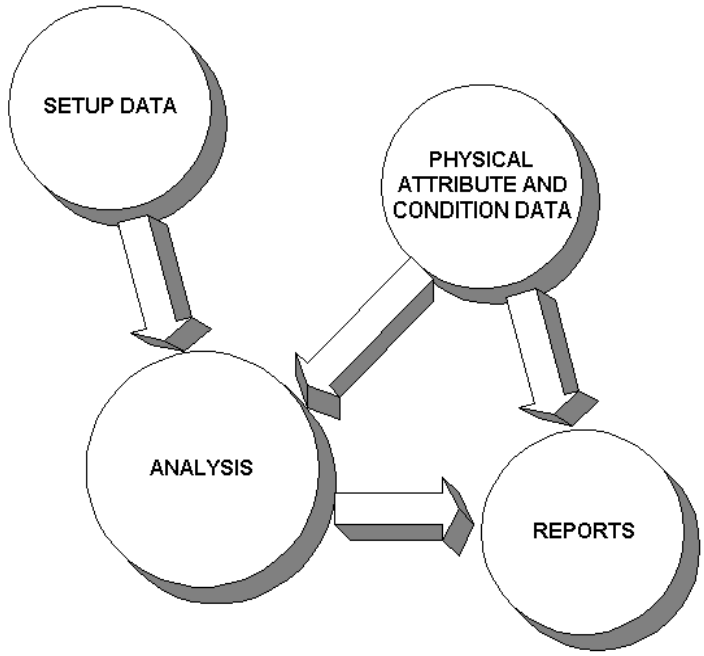


Figure 5.1 Overall Flow of Information

Since pavements and bridges have different physical and condition data and different analysis procedures, the information flow differs for these two types of infrastructure. For pavement management, for example, the flow of information is shown in Figure 5.2.

For bridge management, the analysis is divided into two parts: flagging analysis or a network level selection filter, and the specific project selection procedure. A specific project selection procedure for bridges is further divided into two parts: the M&R project selection and a specific inspection and/or operational restriction activity assignment. Figure 5.3 shows the overall diagram for bridge management.

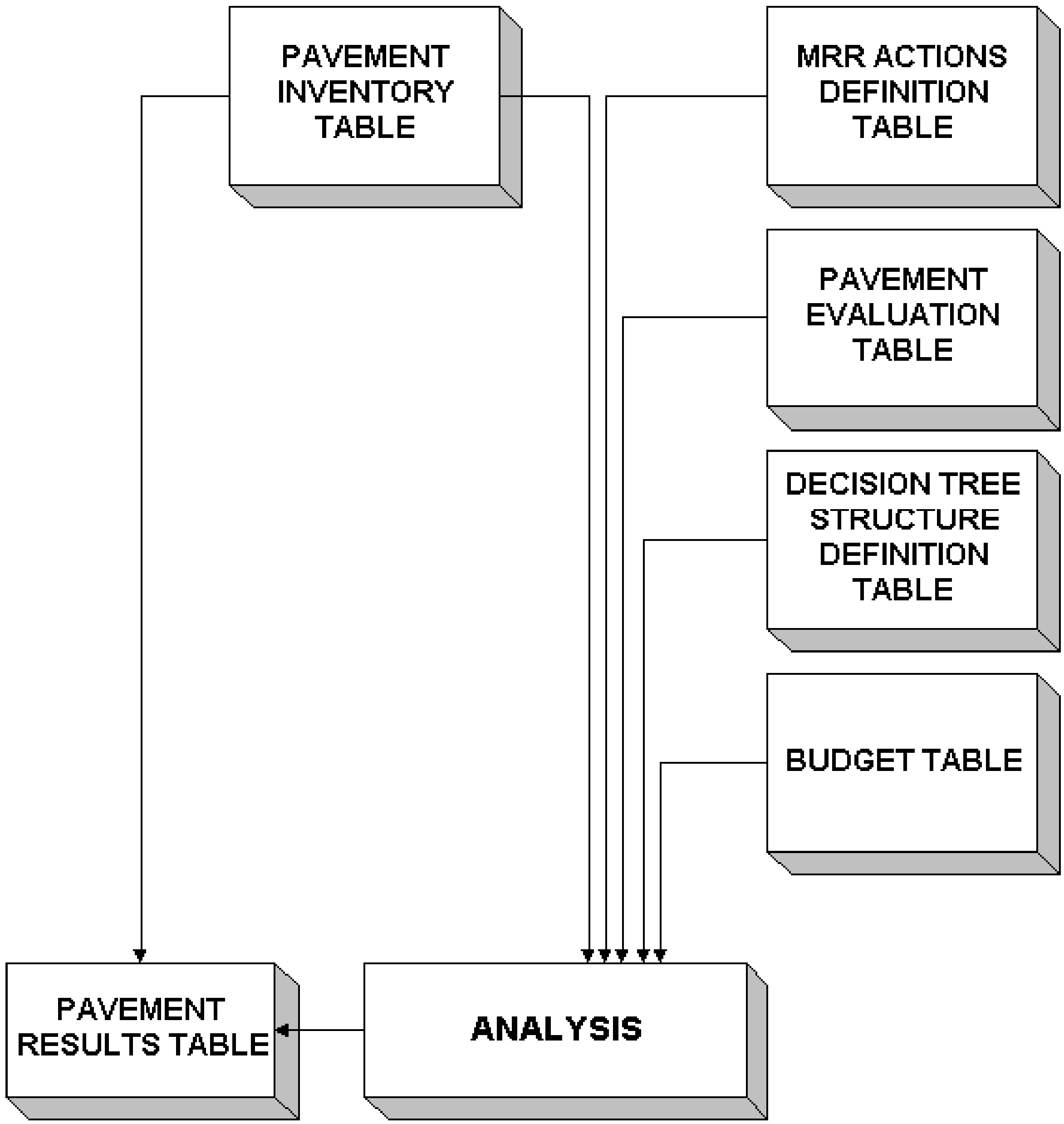


Figure 5.2 Data Flow for Pavement Management Analysis

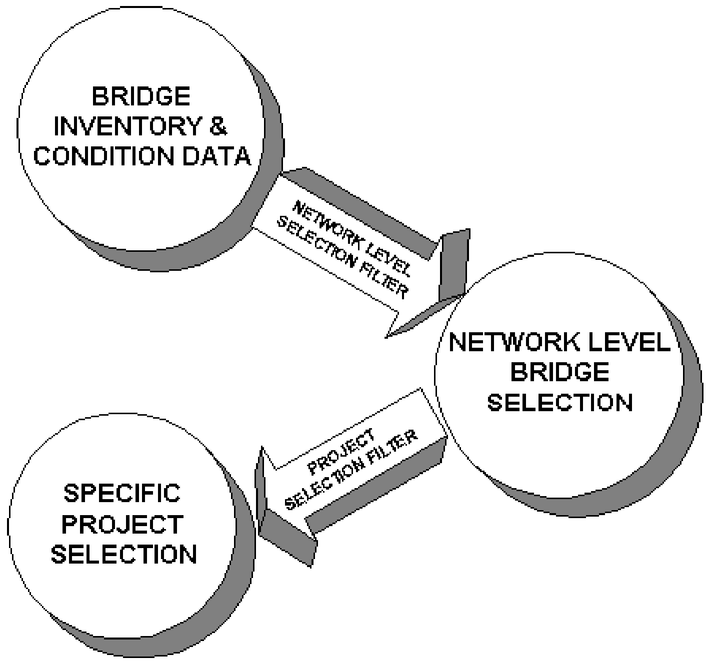


Figure 5.3 Overall Diagram for Bridge Management

Figure 5.4 illustrates the flow of information through the data tables for bridge management part.

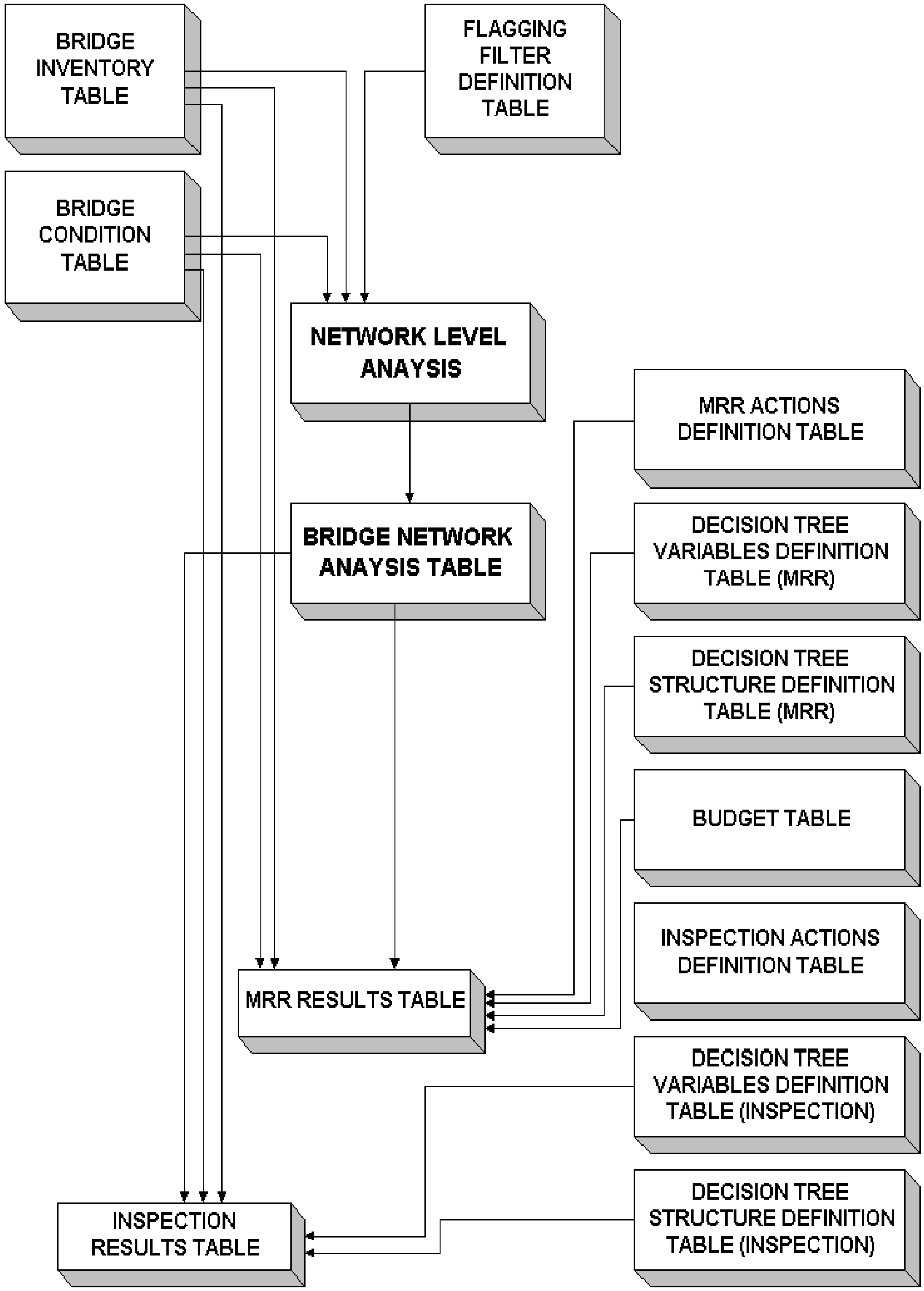


Figure 5.4 Data Flow for Bridge Management Analysis

CHAPTER 6
UIMS SOFTWARE

The Urban Integrated Management System (UIMS) software is a tool that supports the management of city roads and bridges. It helps decision makers in managing bridges and pavements at both the network and project levels. As a computerized management tool, UIMS uses the models and formulations described in previous chapters. This chapter describes the UIMS software structure and provides guidelines for using the software. The example scripts for the original code for the software are provided in Appendix B.

UIMS is supported by a graphical user interface which makes it easier to use. The software is coded using PowerBuilder©, a powerful software package which can create applications that deal with small to medium-sized databases [Power 97]. The final product of UIMS is an executable computer program running under the Windows environment.

6.1 SYSTEM COMPONENTS

The UIMS software is composed of four parts as shown in Figure 6.1: Setup, Database, Analysis, and Reports. The direction of the arrows in Figure 6.1 shows the direction of information flow within UIMS. First, the user defines the maintenance, rehabilitation and replacement (MRR) needs and decision tree structures that are used in analyzing the bridge and pavement infrastructure. This information is combined with the information in the database to perform the analyses. UIMS can produce reports for both the results of these analyses and the raw data itself.

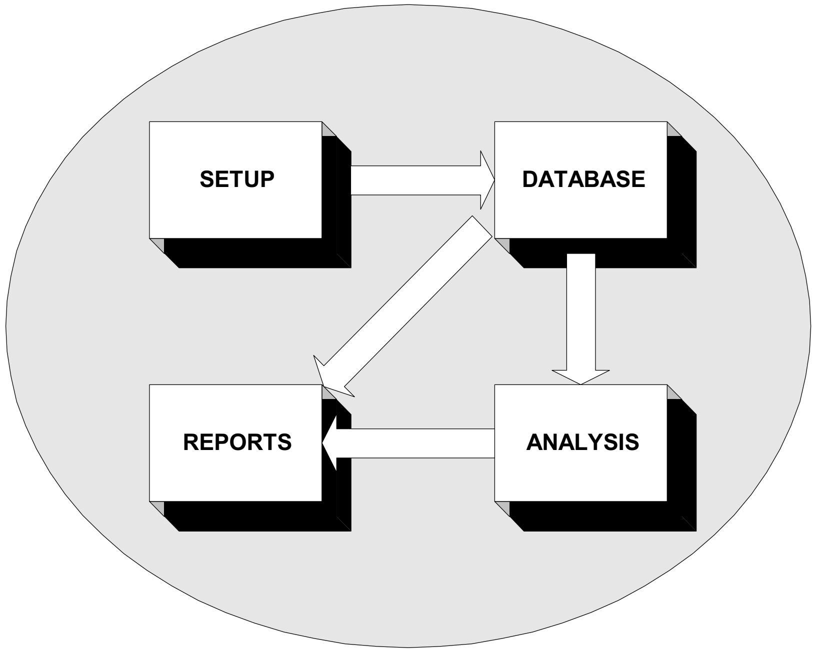


Figure 6.1 System Components of UIMS

6.2 MENU STRUCTURE

The main menu structure of UIMS consists of six main parts: Utilities, Setup, Database, Analysis, Reports, and Help. Each part has its own menu items that perform different functions. Figure 6.2 shows the detailed menu structure of UIMS.

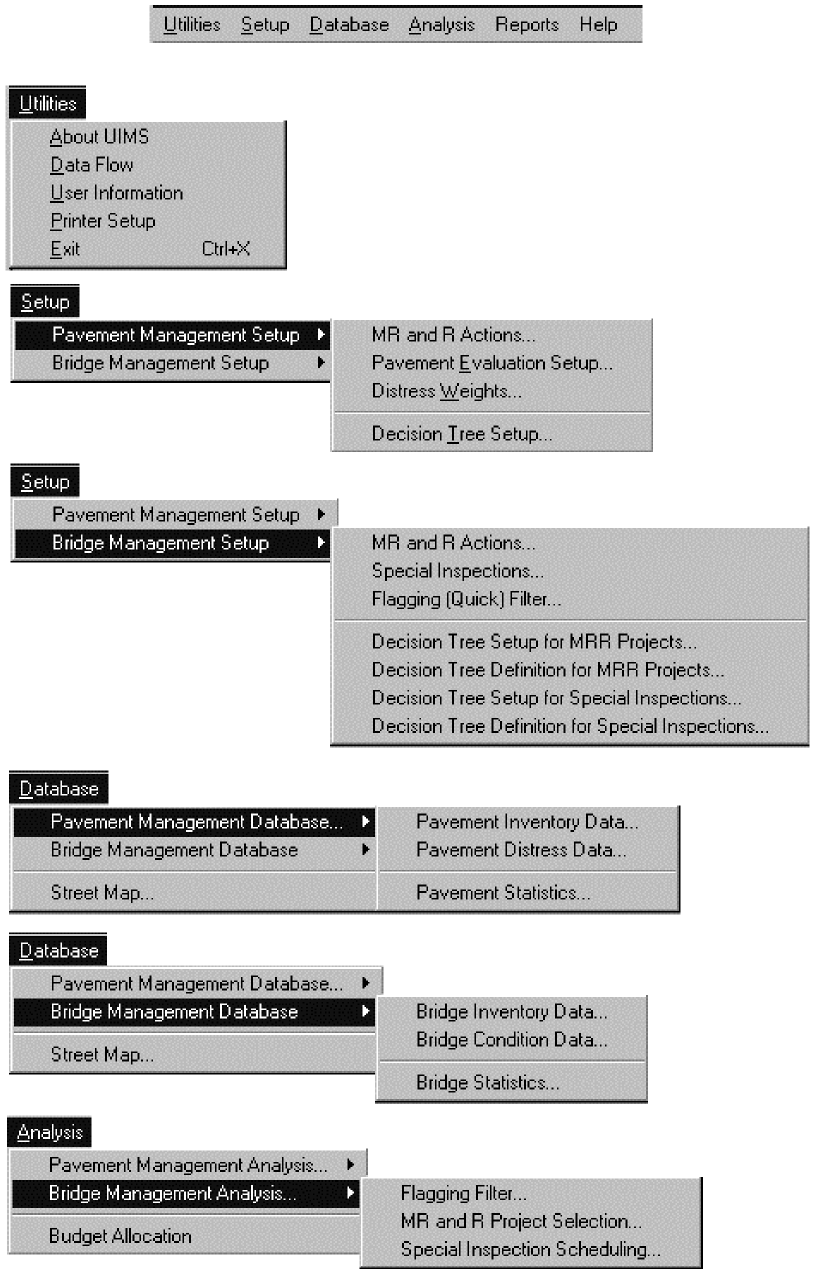


Figure 6.2 Main Menu Structure of UIMS

6.3 USING UIMS

UIMS is user-friendly software that works under the Windows environment. For each function within the software, there is a window and within each window, there are several command buttons. Table 6.1 lists the command buttons in UIMS and describes their functions.

Table 6.1 Command Buttons and Their Functions

Import	Provides for the importing of data from an outside source. If data is available in a text (ASCII) file, UIMS can import that file into its database.
Export	Provides for the saving of data into external files. This is good for back-up purposes. The exported file can be saved in different formats (like ASCII, DBASE, EXCEL, etc.)
Save	Provides for the updating of the database if there are any changes made or new data have been entered. In some windows, it appears as an OK button.
Insert Row	Inserts a new row, or in some cases, a new page to a data window.
Delete Row	Removes a row or page of data from the database.
Filter	Sets a filter expression to view only some portion of the data. The filter does not change what is on the database, it is for display purposes only.
Sort	Sets a sort expression to display the data in a user-defined order.
Apply Decision Tree	This button appears in analysis windows. It brings together the setup information, inventory and condition data and applies the decision trees to assign specific projects to each bridge or pavement management section.
Calculate	Calculates the total costs for each candidate project, and highlights the ones that are within the defined budget.
Sort by Cost	Sorts the projects according to their total costs.

Table 6.1 Command Buttons and Their Functions (Cont.)

Show Pavements	This button appears in the "Street Map" window. It makes the pavement management sections visible on the screen.
Hide Pavements	This button makes the pavement management sections invisible.
Show Bridges	This button appears in the "Street Map" window. It makes the bridges visible on the screen.
Hide Bridges	This button makes the bridges invisible.
Close	Closes the current window. If any changes have been made, the user is prompted whether to save the changes or not by means of a message box.

6.3.1 Window Structures and Functions

6.3.1.1 Getting Started

All the windows in UIMS are accessed via the main window that is shown in Figure 6.3. The UIMS start-up window appears while the software connects to the database. This window is displayed for a few seconds and then disappears. After this point, the UIMS is available to be used with the main window open. By means of the menus, the user selects the portion of the program to use. After each selection a new window shows up in the main window.

6.3.1.2 Utilities Menu

The first menu item in the UIMS is "Utilities". This menu item includes the tools for setting up the general software requirements, including user sign-on and printer setup. As shown in Figure 6.2, the first command under the Utilities menu is called "***About UIMS***". When the user clicks on this command, the UIMS information window appears as shown in Figure 6.4.

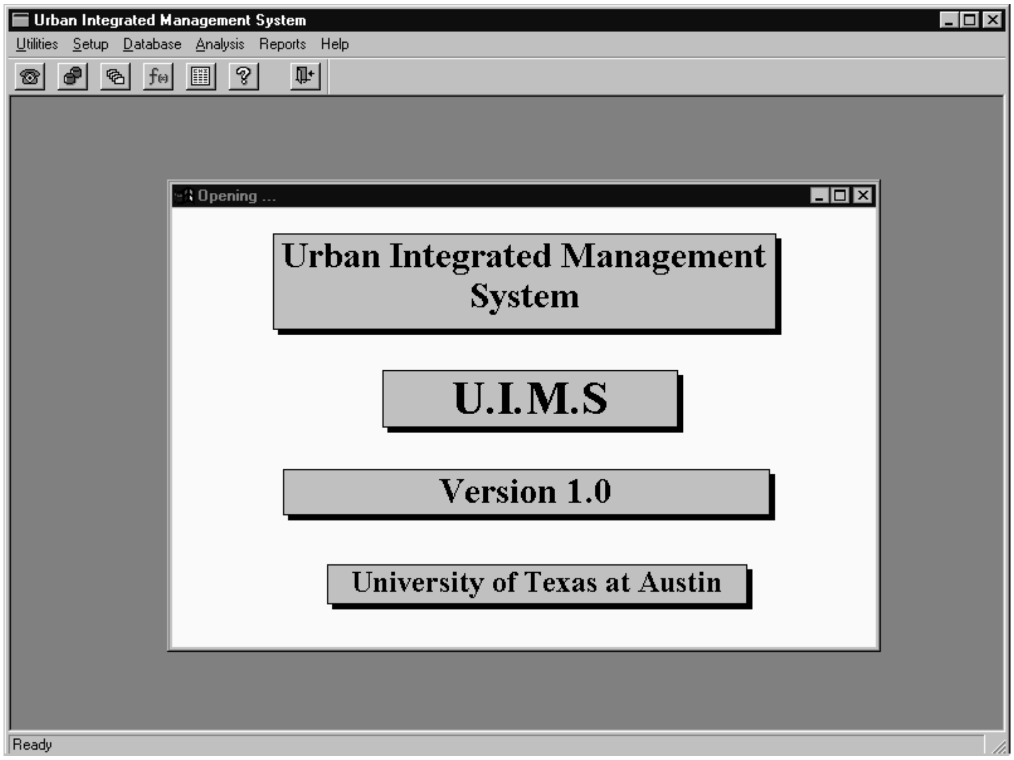


Figure 6.3 UIMS Main Window with Opening Logo

This window is just a display window that gives the user information about the UIMS software; such as the version of the software and the agency for which it has been developed. After reading the information in this window, the user clicks the OK button to close this window and switch back to the main window.

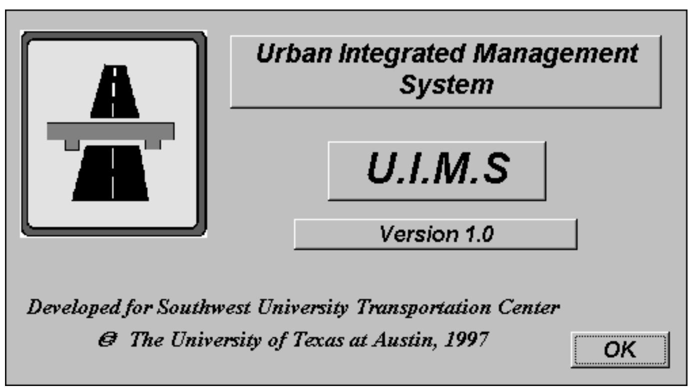


Figure 6.4 "About UIMS" Window

The second command under the Utilities menu item is called "***Data Flow***". This window provides the user quick information about the flow of information within the software. As seen in Figure 6.5, the content of this window looks like a flow diagram. For each box of this diagram, the user can get more information by clicking on the name of that element. As soon as the user clicks on one of the boxes, a pop-up window appears and displays information about the box. For example, Figure 6.5 shows a situation where the user clicked the "Setup Data" box. To remove the information pop-up window, the user can either double click on it, or click on the cross sign (X) at the upper right corner of the window.

Another command in the "Utilities" menu is the "***User Information***". This command activates the user sign-on window shown in Figure 6.6. It includes data related to the user of the software. The first field is for the name of the person who will use the software and perform the analyses. The second field is for the agency, for example the name of the city, or organization that is responsible for the management job.

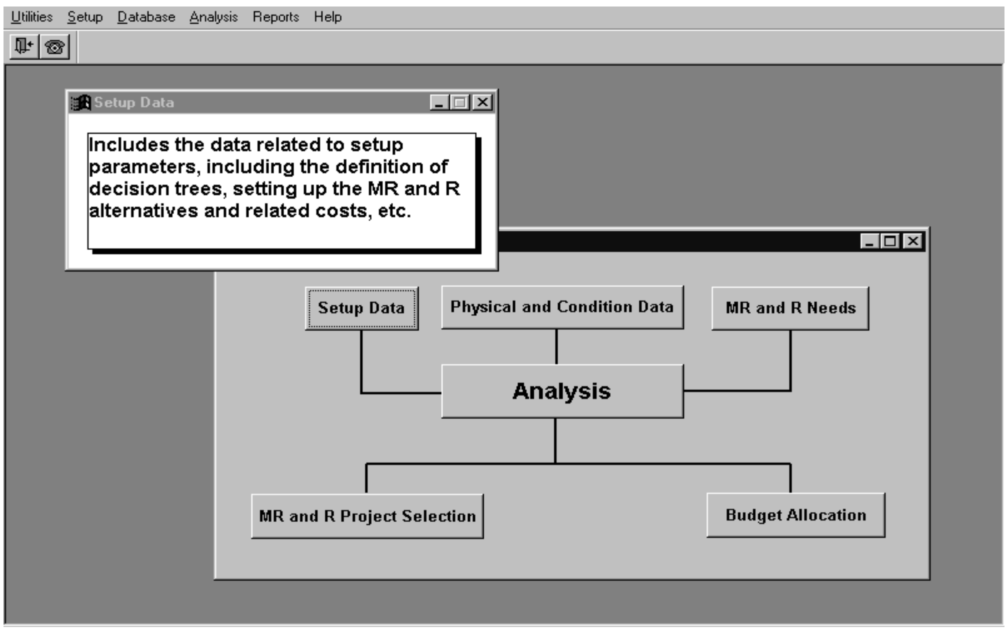


Figure 6.5 "Data Flow" Window with one Pop-up Window Open

The last field is for the date on which the software is being used. After entering or changing the data in this window, the user clicks on the OK button to save the changes and close the window. The information entered in this window appears in the final reports.

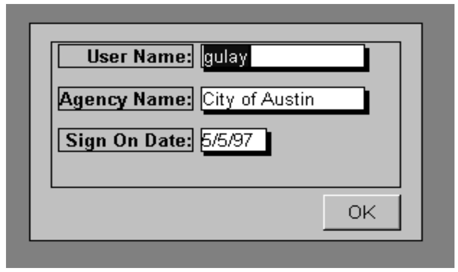


Figure 6.6 User Information Window

In order to print results from the UIMS, it is necessary to define and setup a connected printer. This is done by means of the "***Printer Setup***" command in the Utilities menu. When the user clicks on this command, the printer setup window appears on the screen as shown in Figure 6.7. This window first displays the list of printers that are available (or connected) in the system. The user selects one printer among the list by clicking on the name of the printer. To confirm the selection and close this window, the user should press the OK button. This sets the selected printer with its default setup. In order to change the setup parameters for the selected printer, the "Setup" button is used. After clicking on this button, the properties of the selected printer are displayed on the printer properties window as shown in Figure 6.8 below.

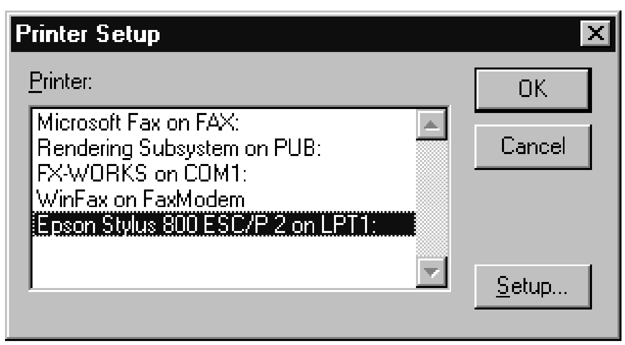


Figure 6.7 Printer Setup Window

Every kind of printing option, such as paper position, graphics printing quality, and paper size can be defined using this window. After making necessary changes, the user should press the OK button to save.

As shown in Figure 6.8, there are several more options related to the printer setup definition. Since UIMS is a windows program and it works under Microsoft Windows©, More information about the printer setup can be obtained from the help file of the Microsoft Windows© while running the program [Windows 95].

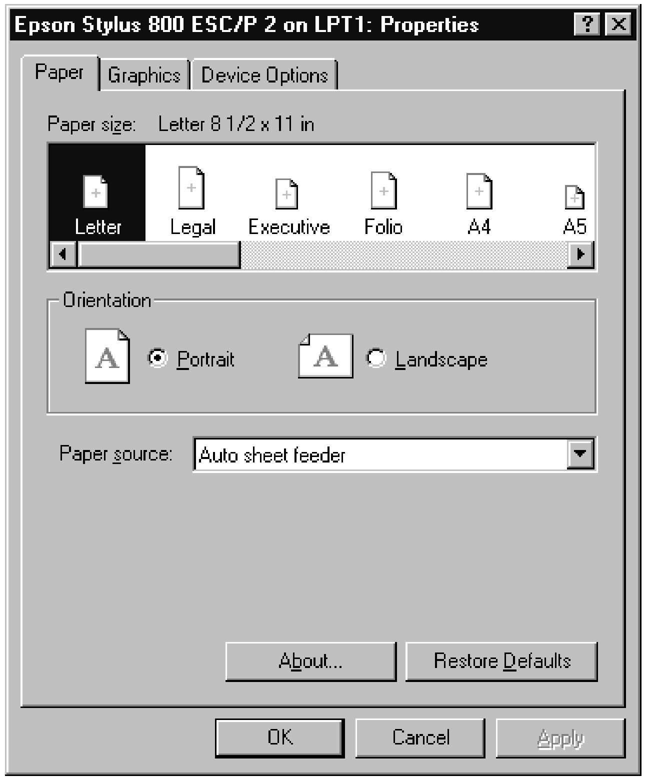


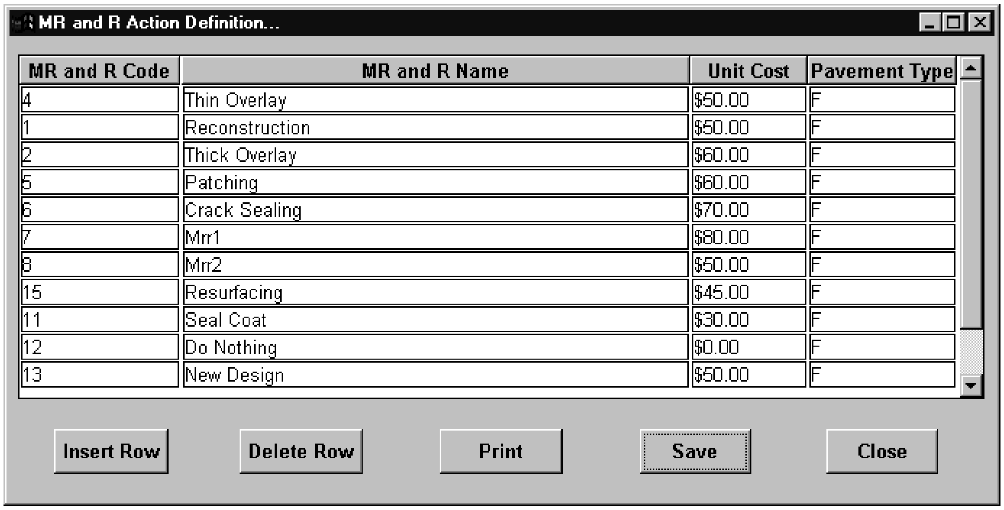
Figure 6.8 Changing Printer Properties

The final command in the Utilities menu is the "***Exit***" command that closes all the windows and exits the UIMS program.

6.3.1.3 Setup Menu

The "Setup" menu is the second menu in the UIMS program and it provides the definition of the necessary information related to analyses. It contains two main parts, one for pavements and one for bridges as shown in Figure 6.2. The setup menu is for the user to define maintenance and rehabilitation actions, the boundaries that separate different levels of different variables (for example, definition of the boundaries for low, medium, and high levels of ADT), and the decision trees that will be used in the analyses.

The first command in the setup menu for either pavement or bridges is "***MR and R Actions***". When the user selects this command, the MRR definition window appears on the screen as shown in Figure 6.9.



MR and R Code	MR and R Name	Unit Cost	Pavement Type
4	Thin Overlay	$50.00	F
1	Reconstruction	$50.00	F
2	Thick Overlay	$60.00	F
5	Patching	$60.00	F
6	Crack Sealing	$70.00	F
7	Mrr1	$80.00	F
8	Mrr2	$50.00	F
15	Resurfacing	$45.00	F
11	Seal Coat	$30.00	F
12	Do Nothing	$0.00	F
13	New Design	$50.00	F

Figure 6.9 MRR Action Definition Window for Pavement Management.

This window provides a tool for the user to input the possible maintenance, rehabilitation and replacement actions that can be applied to pavements and bridges. In order to enter a new MRR action, it is necessary to click the "Insert Row" button that adds an empty row to the window. Then the user fills in the empty fields with the proper data. The first column is the MRR action code, which is a unique number for each action. If the user tries to give a duplicate code number, UIMS does not save it to the database, because this number will be used in further analyses to identify the action name and determine the cost related to each project. The second column is the MRR name which is defined by the user for each project. The user should also define the unit cost for this action on this window. For pavements, there is another column called "Pavement Type" to distinguish between flexible and rigid pavements.

After inputting the proper data, the user should click the "Save" button to update the data tables. Otherwise, if any changes are made and the user clicks the "Close" button without saving the data, UIMS will ask the user whether to save the changes to the database or not. The "Print" button on this window is for printing the data.

Another command for the setup menu is the "***Evaluation Setup***" that activates the Decision Tree Setup as shown in Figure 6.10. In this window, the user enters the lower and upper boundaries for the five levels of pavement condition index, age, and traffic. These values are then used in the analyses part when assigning the MRR projects to the management sections by means of the decision tree. In order to have an understanding about the overall values in the network, the user can press the "View Statistics" button. This button displays the pavement network statistic values as explained later in this chapter.

Some of the fields in this window are in red color and cannot be modified. These fields are common boundaries for two successive levels; for example, the boundary for the "Bad" level for PCI (in Figure 6.10, it is 40) is also the lower boundary for the "Poor" level. When the user changes the content of the "Bad" boundary, the lower boundary of the "Poor" level is updated by the software automatically.

As in every data entry window, after making any changes, the user should click on the "Save" button to save the changes to the database. In order to print the settings, the user should press the "Print" button.

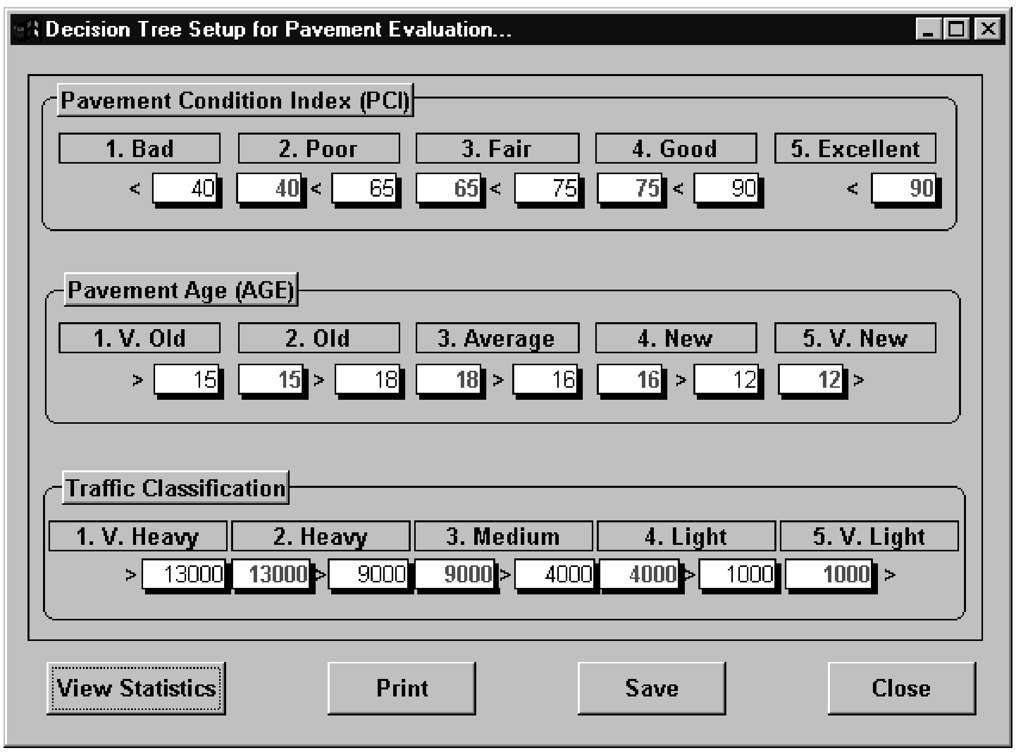


Figure 6.10 Decision Tree Setup for Pavement Evaluation.

The next setup command is the "***Distress Weights***". Using this command, the user can update the weights of different distresses that will be used in a combined distress index. When the Distress Weight window opens, the user can change the new distress weight value for each distress by editing new numbers into the proper places. There are three severity levels for each distress type: low, medium, and high, as shown in Figure 6.11 below.

UIMS allows the user to use 20 different distresses for flexible and rigid pavements. The user may change the name of a distress to a name other than the UIMS defaults. After the distress names and weights are entered, the graph at the bottom portion of the window updates automatically and displays the entered values. This graph displays the distress for each pavement type (for flexible and rigid pavements). Using the radio buttons at the lower right portion of the window, the user can switch between different pavement types. For example if the user clicks on the “Rigid” button, the graph changes and shows the distress weight values for rigid pavements automatically.

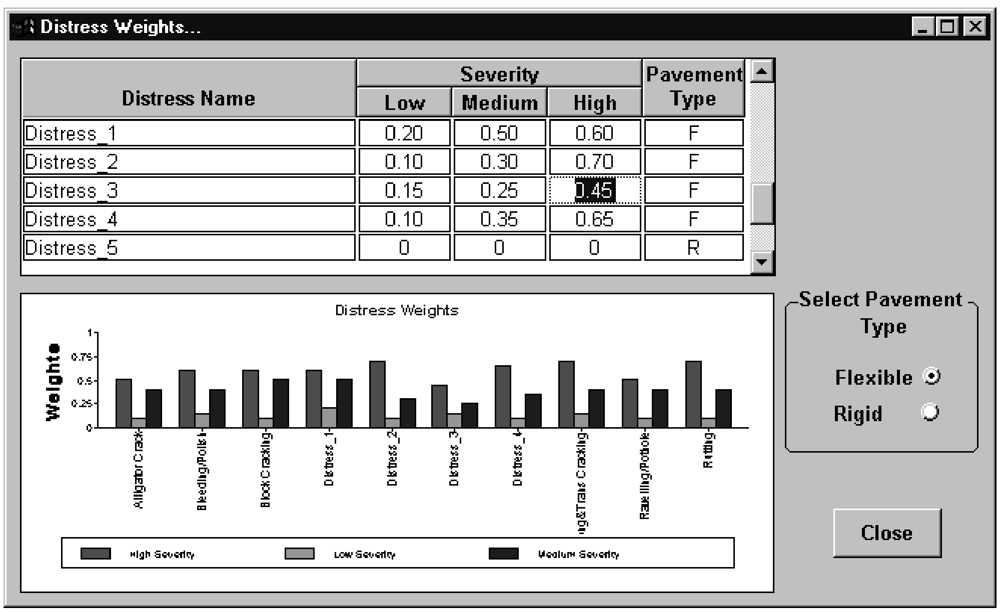


Figure 6.11 Distress Weight Setup.

Another setup menu command is the “***Decision Tree Setup***” used to determine the decision tree structure for pavements and bridges. For example, Figure 6.12 shows a decision tree setup window for pavements. There are three decision variables, PCI, AGE, and ADT, and their levels have already been defined.

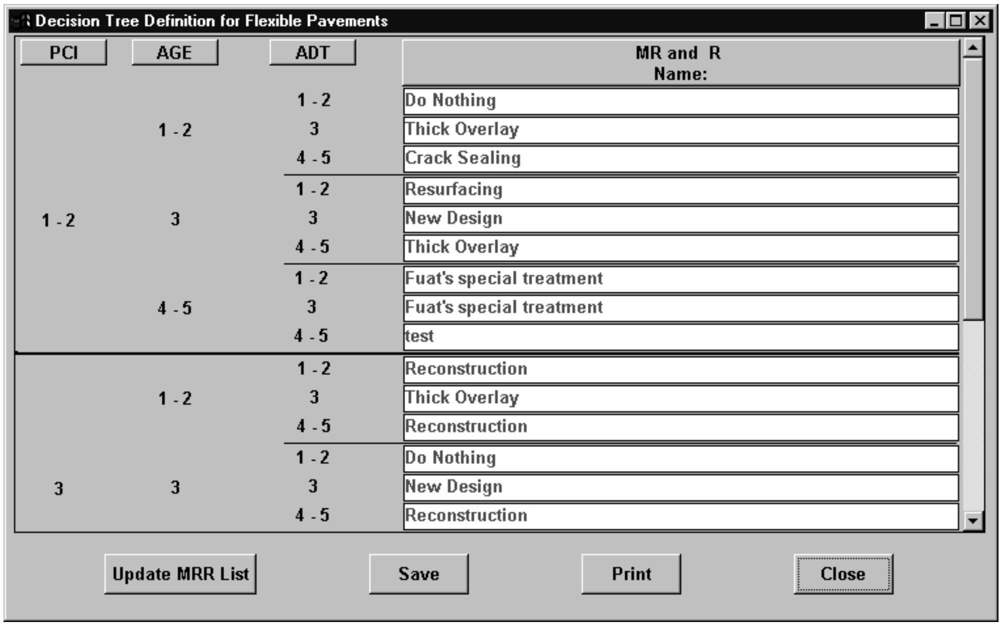


Figure 6.12 Decision Tree Structure for Pavements.

The user is allowed to change the MR and R code in this window, using the MR and R actions that have already been defined in the "***MR and R Actions***" part of the setup menu. To change the MRR action for a specific branch of the tree, the user clicks on the associated MRR name for that branch to choose an action name as shown in Figure 6.13. The user selects a name from this list by clicking on the desired name. For example in Figure 6.13, for PCI, AGE and ADT levels of 1-2 "Resurfacing" action is selected.

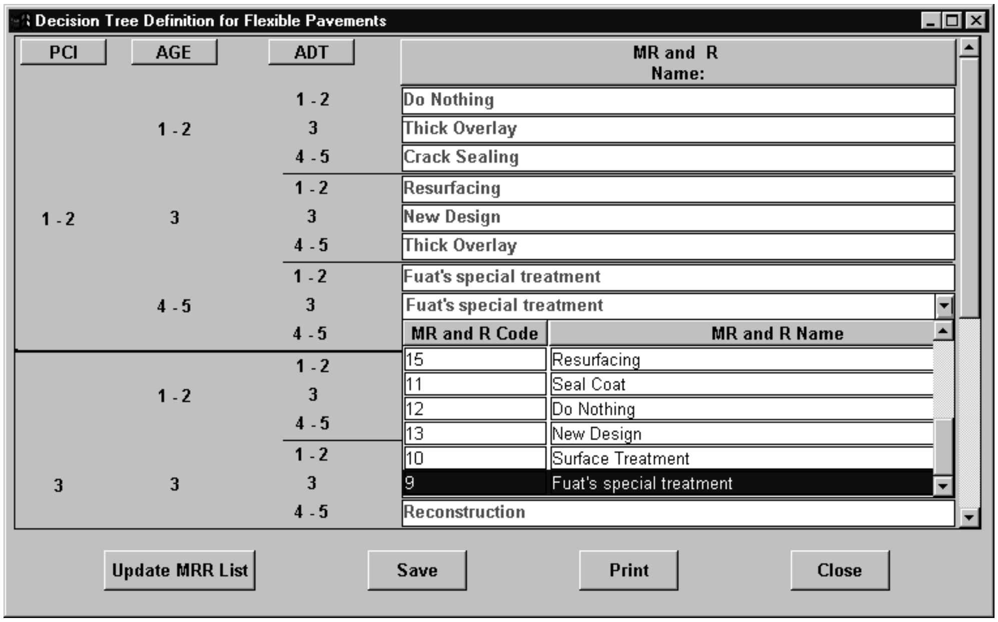


Figure 6.13 Changing MRR Action for one Branch of the Decision Tree

For the bridge management setup, there are some specific commands and one of them is the "***Special Inspections***" command for defining special inspections and operational restrictions that are applicable to bridges. When the user clicks on this command, the special inspections window appears on the screen as shown in Figure 6.14. In terms of inserting, deleting, and editing the data, this window is very similar to the MRR actions definition window. The only difference is that in this window, there is a data field called "Frequency" as opposed to the "Unit Cost" column. The definitions in this window are used in the second decision tree which assigns the inspection and operational activities for bridges.

UIMS also provides a flagging option for the user to view the bridges with certain deficiencies. This option can also be used as a network level filter to reduce the number of candidate bridges for further analyses.

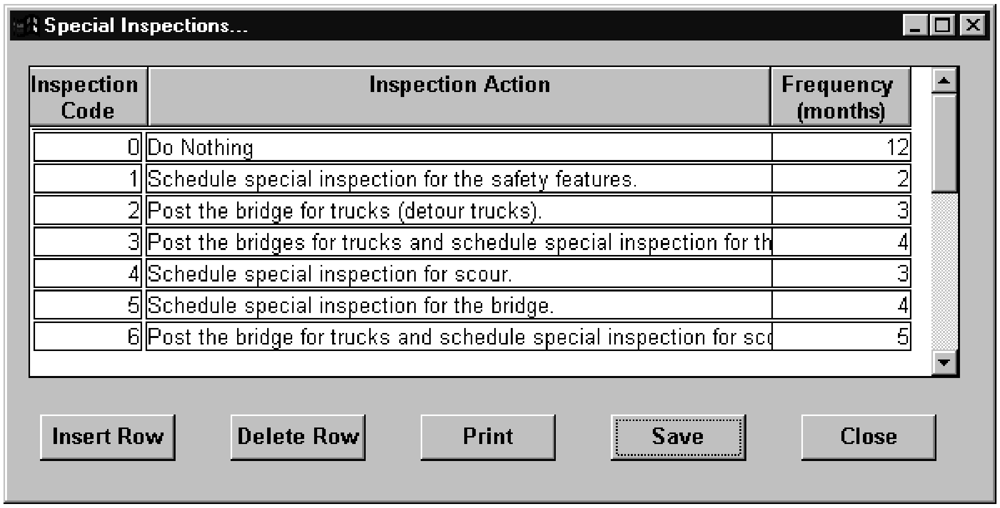


Figure 6.14 Special Inspection Definition Window

The flagging filter function is provided by the "***Flagging (Quick) Filter***" command of the setup menu. When the user selects this command, the flagging filter window appears on the screen as shown in Figure 6.15. On this window the user can enter values for different variables. The variables included are: the historical significance code, bridge length, ADT, average bridge condition, safety features, scour critical code, deck width, and allowable load class for the bridge. The values entered on this window are used as a quick reference to identify the critical bridges according to one of these variables. The bridges that are selected by this filter can also be moved to the analysis part. In this respect, this flagging filtering option can also be used as a network level bridge selection tool in UIMS.

6.3.1.4 Database Menu

The database menu is the third menu item in the UIMS main window. It includes the commands for basic data entry and data manipulation functions for both pavements and bridges. The database menu also has two parts as shown in Figure 6.2: the pavement management database, and the bridge management database.

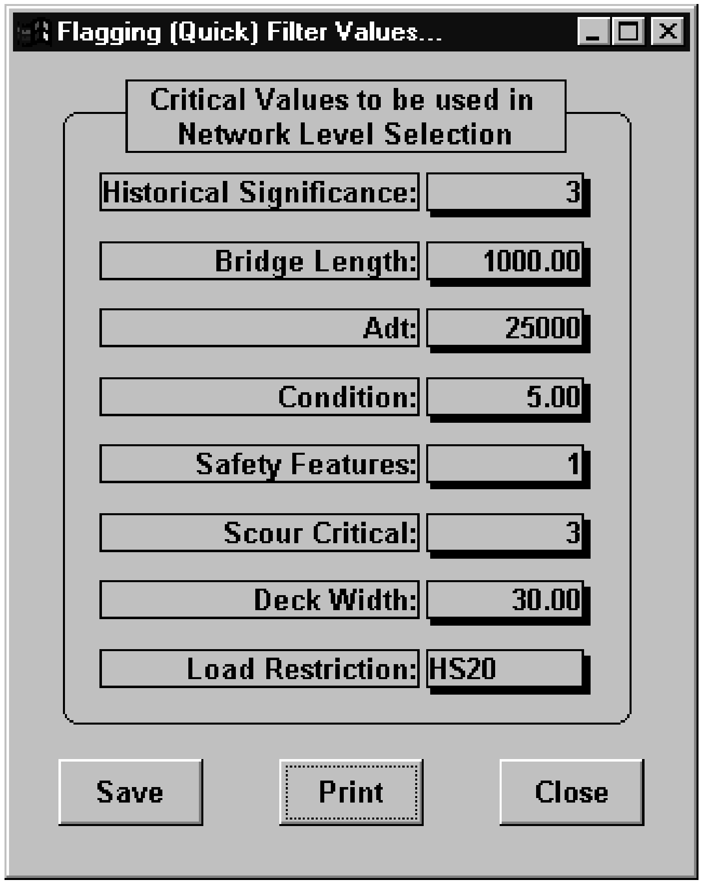


Figure 6.15 Flagging (Quick) Filter Values

The first command in the pavement management part of the "Database" menu is called "***Inventory Data***". This command activates the inventory window for pavements or bridges which provides functions like editing, viewing, or deleting inventory data. Figure 6.16 shows the inventory window for pavements.

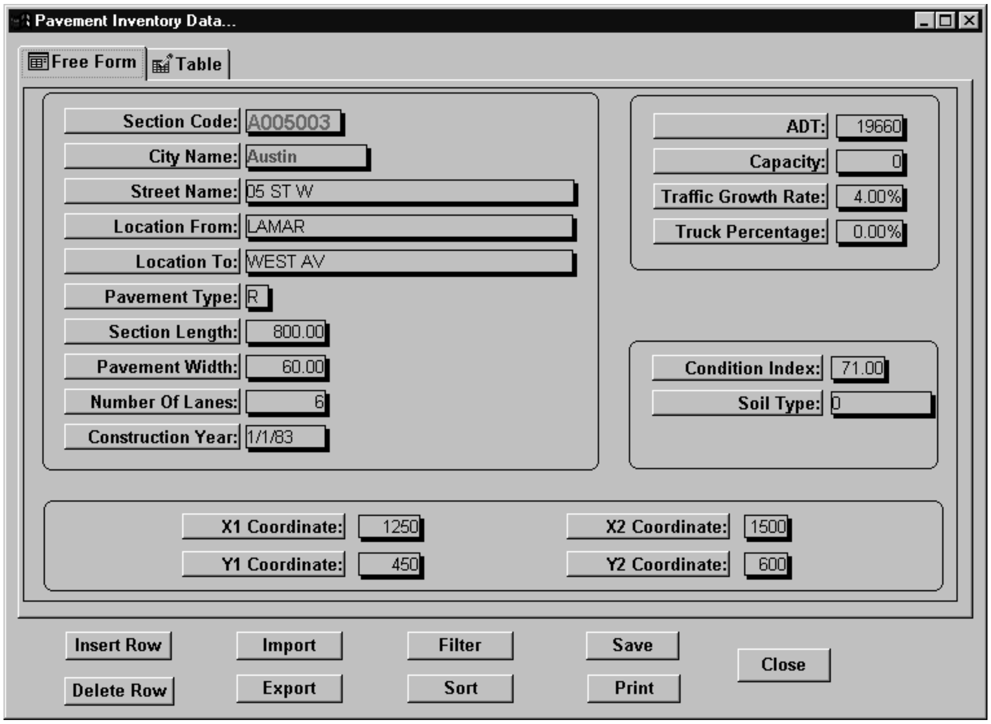
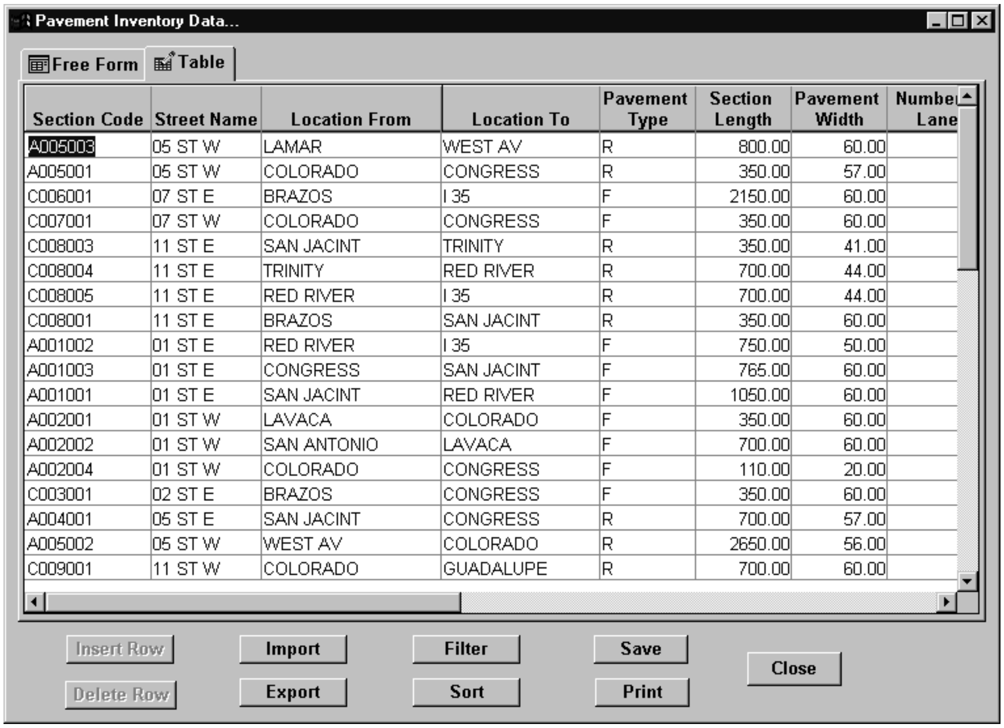


Figure 6.16 Inventory Data Window for Pavements ("Free Form")

As shown in Figure 6.16, there are two tab buttons for switching between the "Free Form" and "Table" data window formats. The "Free Form" format is good for entering new data into the database, because the user can see all the data items in one window. The "Table" form, shown in Figure 6.17, is good for data manipulations such as sorting and filtering. When the inventory data window first opens, the default is the "Free Form" format.

There are several function buttons on the inventory data window. The first one of these buttons is the "Insert Row" button for inserting an empty new data entry page. This button is "Enabled" if the "Free Form" format is selected. In other words, in order to insert new data, the user should select the "Free Form".



Section Code	Street Name	Location From	Location To	Pavement Type	Section Length	Pavement Width	Number Lane
A005003	05 ST W	LAMAR	WEST AV	R	800.00	60.00	
A005001	05 ST W	COLORADO	CONGRESS	R	350.00	57.00	
C006001	07 ST E	BRAZOS	I 35	F	2150.00	60.00	
C007001	07 ST W	COLORADO	CONGRESS	F	350.00	60.00	
C008003	11 ST E	SAN JACINT	TRINITY	R	350.00	41.00	
C008004	11 ST E	TRINITY	RED RIVER	R	700.00	44.00	
C008005	11 ST E	RED RIVER	I 35	R	700.00	44.00	
C008001	11 ST E	BRAZOS	SAN JACINT	R	350.00	60.00	
A001002	01 ST E	RED RIVER	I 35	F	750.00	50.00	
A001003	01 ST E	CONGRESS	SAN JACINT	F	765.00	60.00	
A001001	01 ST E	SAN JACINT	RED RIVER	F	1050.00	60.00	
A002001	01 ST W	LAVACA	COLORADO	F	350.00	60.00	
A002002	01 ST W	SAN ANTONIO	LAVACA	F	700.00	60.00	
A002004	01 ST W	COLORADO	CONGRESS	F	110.00	20.00	
C003001	02 ST E	BRAZOS	CONGRESS	F	350.00	60.00	
A004001	05 ST E	SAN JACINT	CONGRESS	R	700.00	57.00	
A005002	05 ST W	WEST AV	COLORADO	R	2650.00	56.00	
C009001	11 ST W	COLORADO	GUADALUPE	R	700.00	60.00	

Figure 6.17 "Table" Format for Inventory Data Window

The second button is the "Delete Row" button. It is for removing a record from the database. Since deleting a record can be a risky operation, UIMS prompts the user to confirm the deleting operation by means of the message box shown in Figure 6.18. If the user clicks "Yes", the record is deleted.

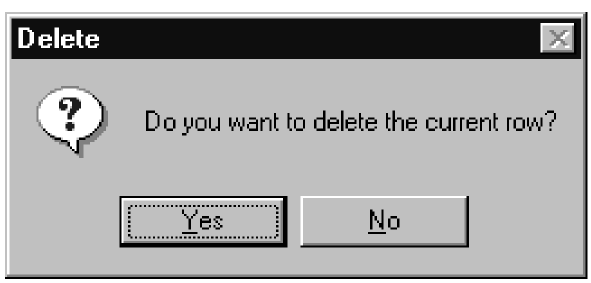


Figure 6.18 "Delete Row" Message Box

In order to make the changes permanent to the database, the user should press the "Save" button after inserting or deleting a record.

UIMS provides functions for importing and exporting data which are convenient for data backup purposes. It also saves considerable time on data entry if there are data available from an external data source. When the user presses the “Import” button, the “Select Import File” window appears as shown in Figure 6.19. The import option allows the user to import data from text or dbase files.

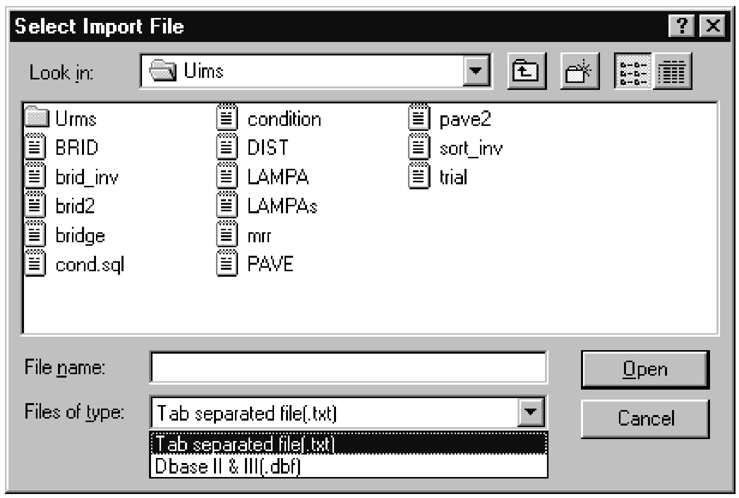


Figure 6.19 Select Import File Window

After locating the file to be imported, the user should press the “Open” button to import the file into the database. To complete the operation, the user should press the “Save” button after the import is finished. Otherwise, imported data is not saved into the database.

To save the existing data into an external file, the user should use the “Export” button. This activates the “Save As” window as shown in Figure 6.20. Options for the export file type include: ASCII, dbase, Excel, Lotus, etc. After determining where to save the file, file name, and type of the file, the user should press “Save” to export the file.

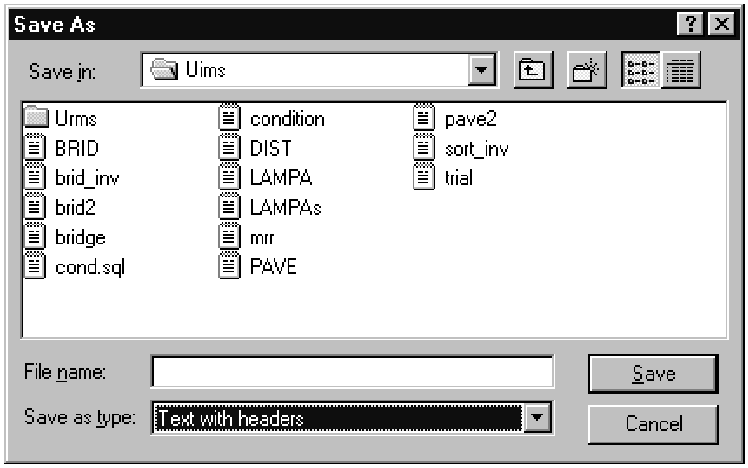


Figure 6.20 Save As Window

Another button is the “Filter” button that can be used to perform manipulations with the current data for display purposes. Using this button, the user can define various kinds of expressions as the filtering criteria. Figure 6.21 shows the “Specify Filter” window. After the filtering expression is specified, pressing the “OK” button displays the portion of the data that satisfies the defined filter. On the “Specify Filter” window, the user can select different functions from the function list and can select the column to be used in filtering from the column list. After setting a filter expression, the user can press the “Verify” button to check if the defined filter is appropriate or not. Logical operators such as “and” and “or” can also be used in defining the filter criteria. For example, Figure 6.21 shows a criteria expression for pavement sections with less than 35 feet width. Figure 6. 22 shows the result of this filtering operation.

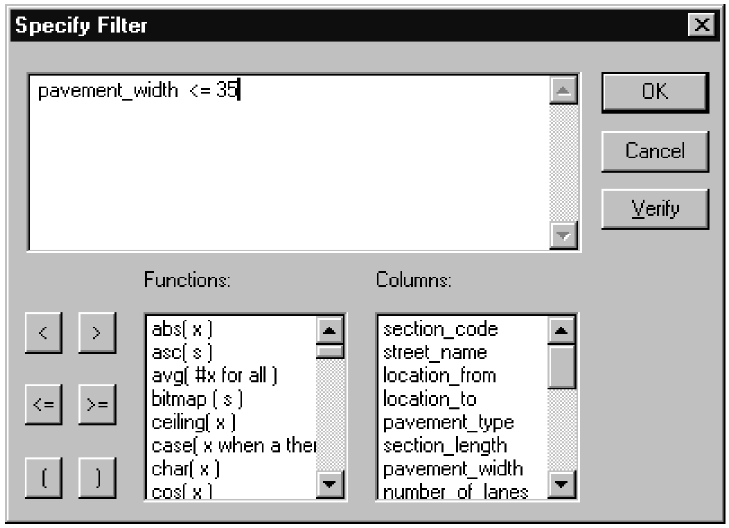


Figure 6.21 Specify Filter Window

The "Sort" button can be used to reorganize the data into a user defined order. The user can select the columns to be used for sorting by means of the "Specify Sort Columns" window shown in Figure 6.23. The "Source Data" portion of this window displays the list of columns in the database that can be used for sorting. The user can select any column(s) from this list by dragging, and putting them into the "Columns" area using the mouse. The selected column values can be used in either ascending or descending order by checking the "Ascending" box.

Figure 6.23 illustrates an example where the user selected two columns as sorting criteria. In this example, the inventory data will be sorted first by "section_code", in ascending order, then by "pavement_width" in descending order. Similar to the "Filter" option, the user can define a sorting criteria based on arithmetical expressions interactively by double clicking the column name.

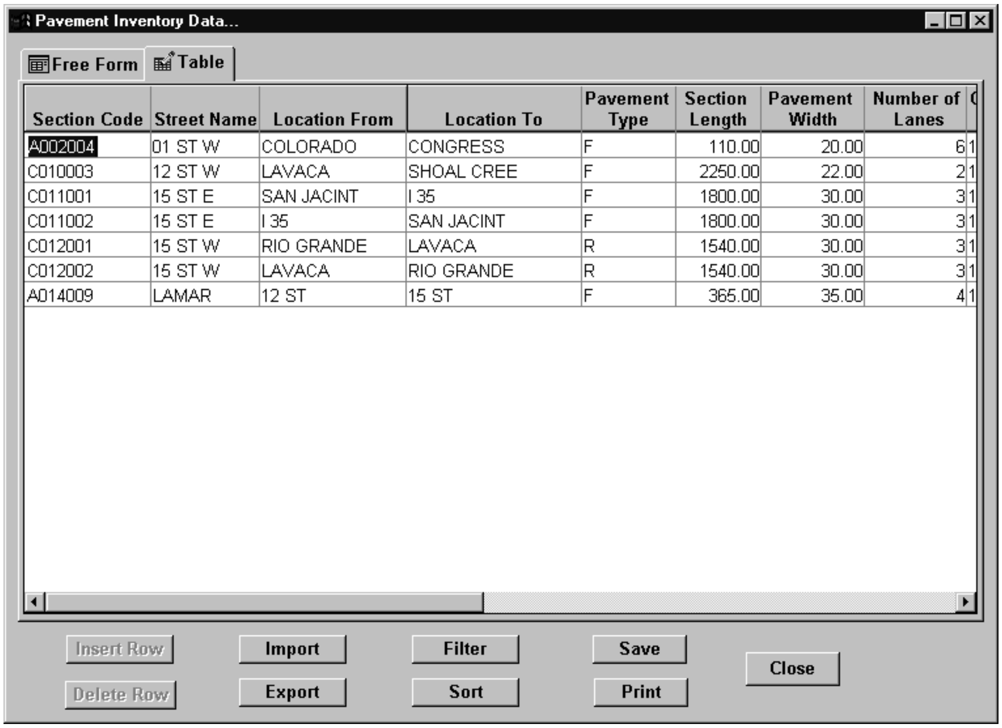


Figure 6.22 Result of Filter for Pavement Inventory Data

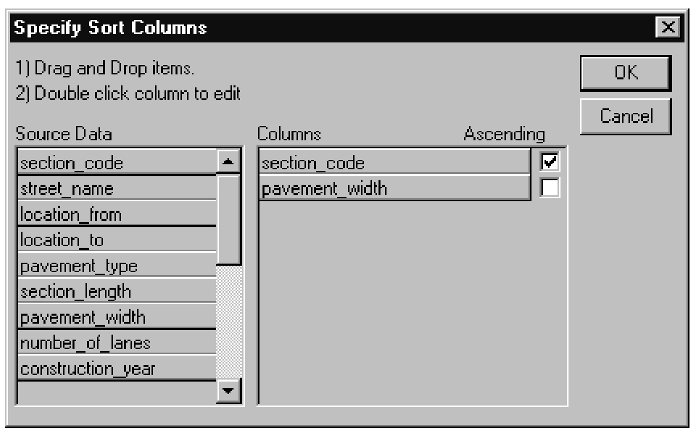


Figure 6.23 Specify Sort Columns Window

The next command in the database menu is the “Distress Data” used to enter and display the distress data for the pavement management sections. A similar command called “Bridge Condition Data” exists for bridges. As shown in Figure 6.24, this window consists of two parts. The upper part is a data table that provides distress data entry for each management section. The lower part is a graph that displays the entered distress values for each section. The x-axis on this graph represents the management sections along a street, and the y-axis represents the distress value for each management section.

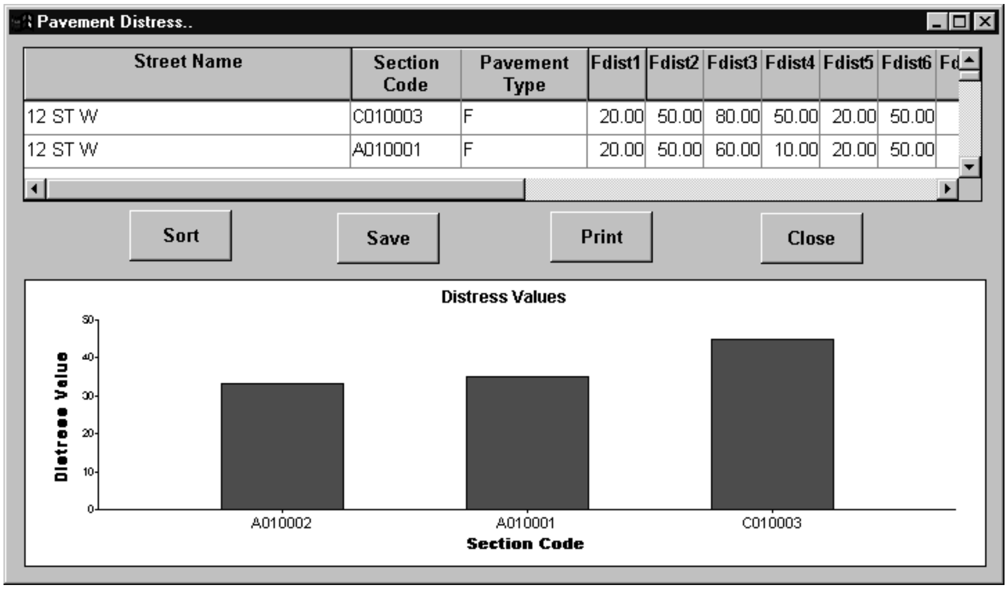


Figure 6.24 Pavement Distress Data Window

To see the plot of distress data for different streets, the user should click on the name of the street from the upper data window. For example, to see the distress distribution for street "12 ST. W", the user just needs to click on a row that has this street's data. The graph will be updated automatically after the selection.

The next command under the database menu is the "***Statistics***" command. This command activates the "Statistics" window that provides basic statistical information about the pavements or bridges. Figure 6.25 shows the statistics window for pavements. There are three parts in this window: a data window that display the average values of ADT, AGE and PCI, a graph selection part, and a graph that displays the distribution for the selected variable.

When the "Statistics" window first opens, the graph part is empty. The user can select the variable for which the graph is to be displayed using the radio buttons within the "Select Graph" area. The graph displays the counts for each variable within a certain interval. The information on this window allows the user to make an overall evaluation of the pavement network as a preliminary guideline in selecting the critical values for the "Pavement Evaluation Setup".

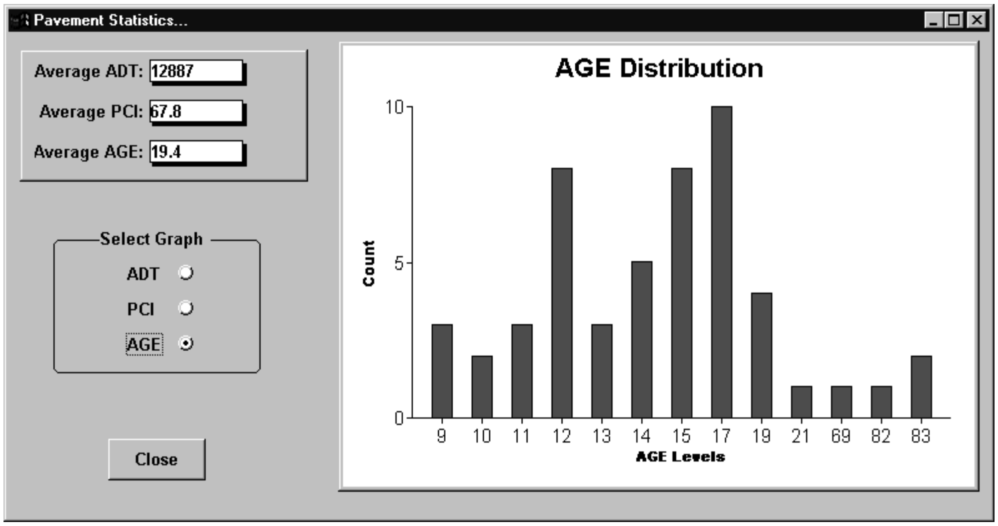


Figure 6.25 Pavement Statistics Window (for AGE)

UIMS allows the user to see the pavements and bridges on a map by means of the "***Street Map***" command of the Database menu. This command activates the street map window shown in Figure 6.26 where the street map is plotted using the pavement and bridge inventory data. This window contains three main parts: the street map display window, the pavement inventory table, and the bridge inventory table.

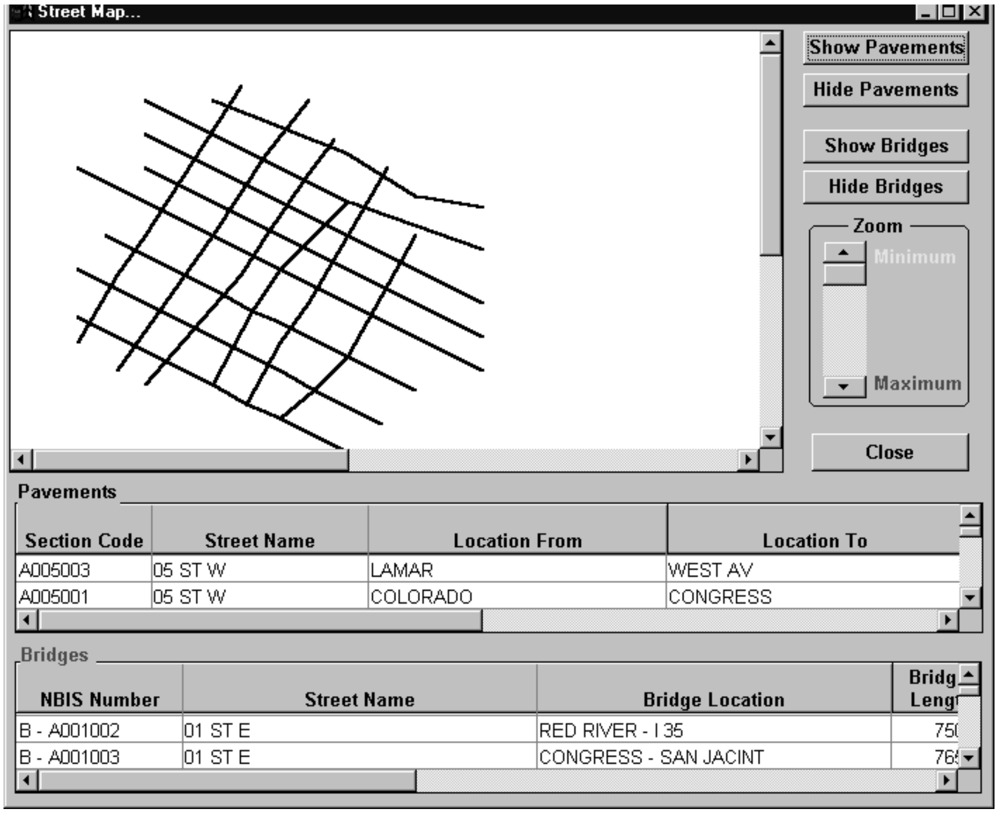


Figure 6.26 Street Map Window (with pavements only)

When first opened, the street map display window is empty. The user can select the type of infrastructure to be displayed by pressing "Show Pavements" or "Show Bridges" buttons. After pressing one of the two buttons, the map display window shows the street map for the selected infrastructure. Figure 6.26 shows the situation where the user pressed the "Show Pavements" button. If the user presses the "Show Bridges" button while the map displays the pavements, the map displays both pavements and bridges. Similarly, it is also possible to hide any of these infrastructures using the "Hide Pavements" or "Hide Bridges" buttons.

The data windows at the bottom part of the street map window can be used for an individual pavement section or bridge to be highlighted in the map. For example, if the user clicks on the first management section from the "Pavements" table, the line segment corresponding to this section on the map changes to the red color. This allows the user to see visually the management section on the map and related information about that section simultaneously. The same function is also applicable for bridges.

Another feature of the street map window is the zooming capability. The user can zoom in and out of the street map using the “Zoom” scroll bar. Zooming is at maximum when the scroll bar is at the lowest position, and minimum at the highest position. Figure 6.27 shows a zoomed segment of the street map.

6.3.1.5 Analysis Menu

The analysis menu is the fourth menu item in the UIMS main window, and it includes the commands that provide pavement and bridge management analyses, such as the specific project selection and the budget allocation. The windows under the analysis menu combines setup, inventory and condition data to analyze the infrastructure network. There are two groups of analysis functions under the analysis menu.

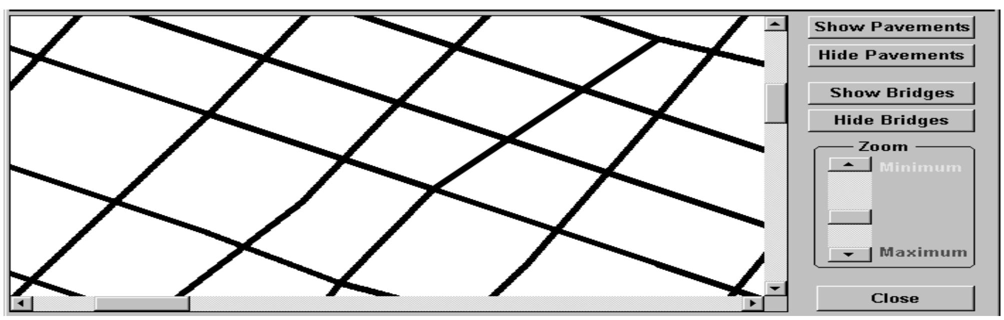


Figure 6.27 Street Map Window with Zooming

6.3.1.5.1 Pavement Management Analysis

There is only one command under this group called, “***Pavement MRR Analysis***”. This command activates the Pavement MRR Analysis window shown in Figure 6.28.

There are two tables on the “Pavement MRR Analysis” window, one is the pavement inventory table (at the top), and the other is the pavement analysis results table. When the user clicks on the “Apply Decision Tree” button, UIMS uses the decision tree setup information and pavement inventory data to select the MRR projects for each pavement management section. “Filter” and “Sort” buttons can be used if the user wants to introduce a filter prior to the selection or wants to rank the selected projects with a user defined criteria. For example, if the user wants to evaluate the pavement sections with 35 ft or less pavement width, a filter expression can be written using the “Filter” button. After the filter is set, UIMS updates the analysis results table for the filtered sections.

In the analysis summaries window, the section code, the MRR action to be applied, the unit cost of the MRR action, the project cost, and the total cost values are displayed. To distribute the available budget (which is usually less than the total cost), the “Calculate” button can be used. The user enters the available budget into the budget field and presses the “Calculate” button. This distributes the available budget among the projects, starting from the top of the list until it ends as shown in Figure 6.28. For this reason, a proper sorting should be provided prior to the budget allocation.

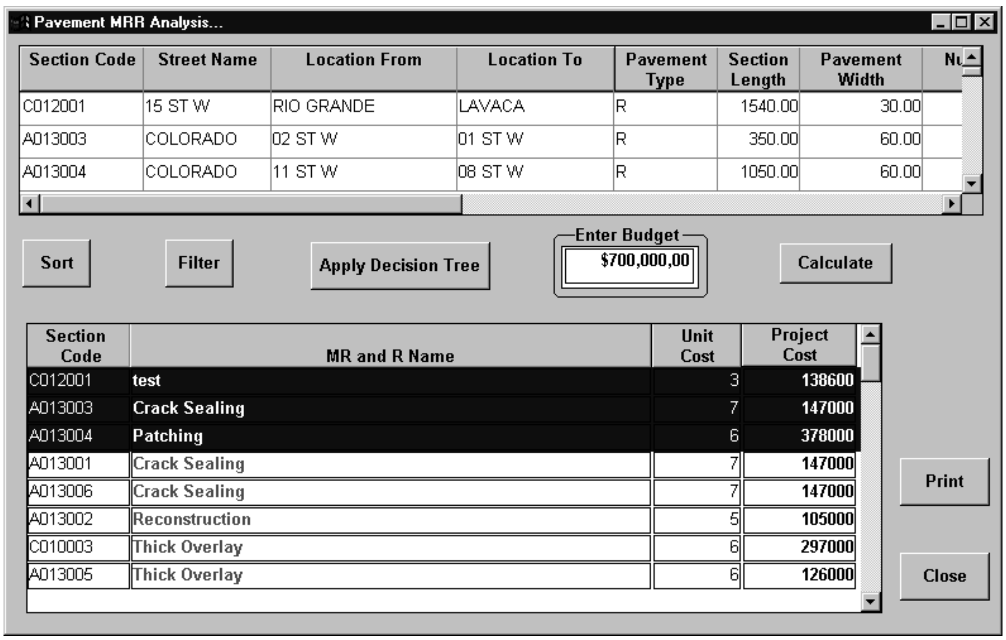


Figure 6.28 Pavement MRR Analysis Results

6.3.1.5.2 Bridge Management Analysis

The bridge management analysis part of the analysis menu includes the commands necessary to analyze the bridge network. The structure of the bridge management analysis part is different from that of pavement management analysis. Specifically, it includes a flagging analysis, a bridge MRR analysis, and a special inspection analysis.

The first command under this group is called the "***Flagging Filter***" which provides a quick filtering mechanism for bridges. When the user selects this command, it activates the "Flagging Filter" window as shown in Figure 6.29 below. This window includes two tables: the inventory table (at top) and the flagging results table. There are also several radio buttons for selecting the filtering variable.

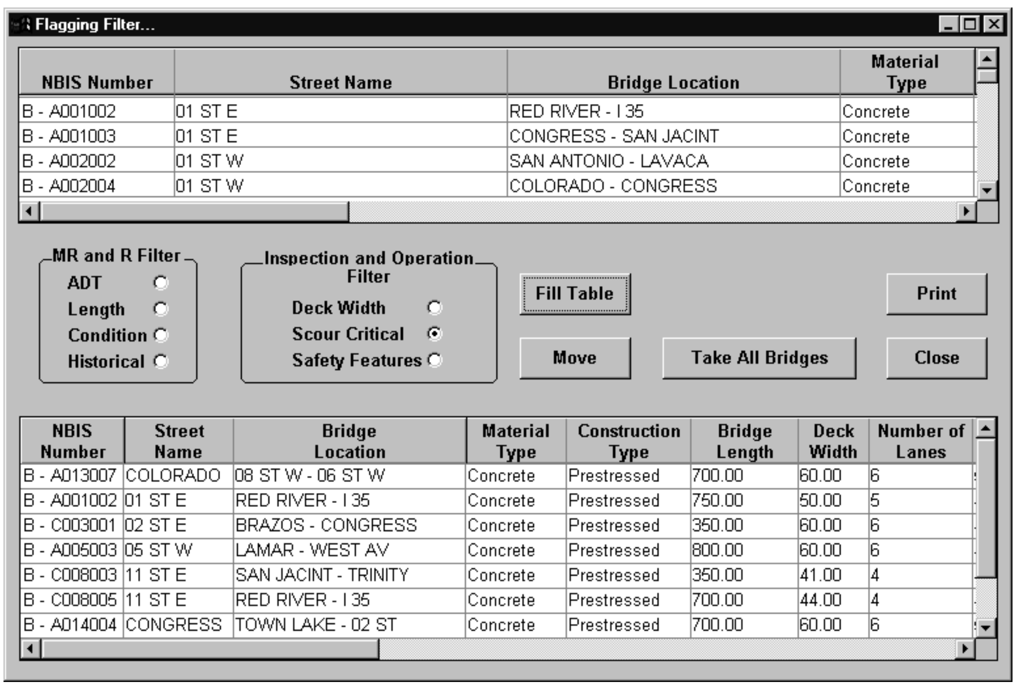


Figure 6.29 Flagging Filter Window

The working principle for the window is fairly simple. The user selects the filtering variable and presses the "Fill Table" button. This action extracts the bridges that meet the filtering criteria of the selected variable. For example, the user can view the scour critical bridges on the bridge network by selecting the "Scour Critical" variable and pressing the "Fill Table" button.

If the user wants to assign specific MRR projects and special inspection actions to the filtered bridges on this window, it is necessary to press the “Move” button that takes the filtered bridges to the next analysis window. If the user does not intend to apply a filtering prior to further analysis, the “Take All Bridges” button should be used.

The next command is the “***MR and R Project Selection***” command which has similar functions as the “***Pavement MRR Analysis***” command. The only difference is that bridge projects are selected instead of pavement projects. Figure 6.30 shows the MRR project assignment window and selected projects for bridges.

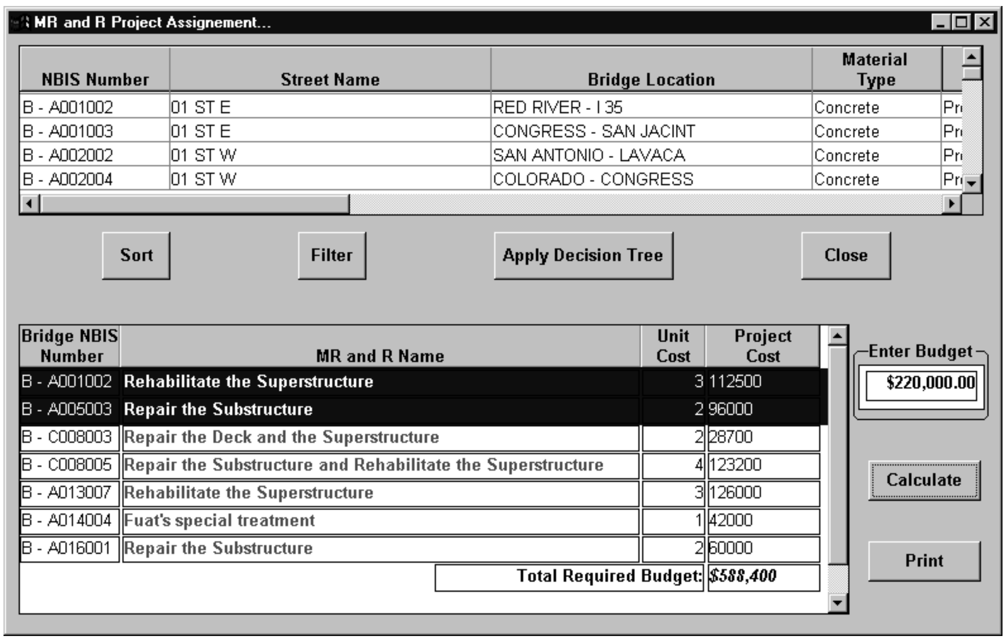


Figure 6.30 MRR Project Selection for Bridges

The last command is the ***“Special Inspection Scheduling”*** command used to select the bridges that require special inspection or operational restrictions. It is similar to the MRR project assignment window, the only difference being that instead of calculating cost, the next inspection date is determined by the software as shown in Figure 6.31.

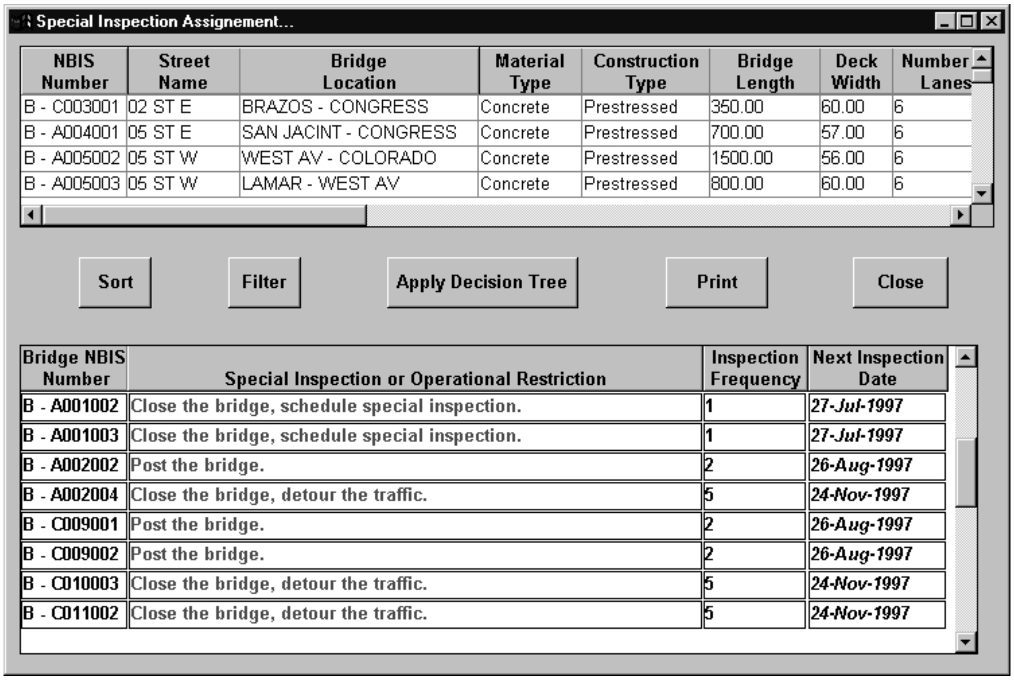


Figure 6.31 Special Inspection Scheduling Window for Bridges

6.3.1.6 Reports Menu

The “Reports” menu consists of a single command that activates the report selection window shown in Figure 6.32. UIMS allows the user to generate predefined reports from the analyses performed. Once the report selection window is opened, the user can view or print a report simply by clicking on the name of the report and pressing the “Show” button.

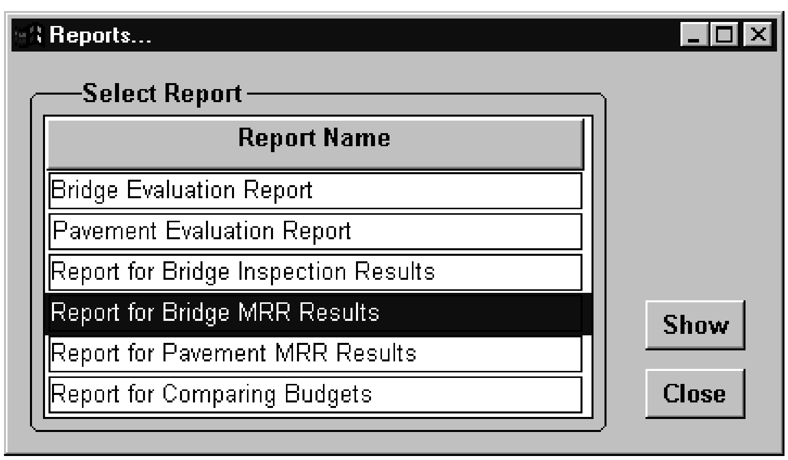


Figure 6.32 Report Selection Window

The “Show” button activates the report viewing window as shown in Figure 6.33. For example, if the user selects the “Report for Bridge MRR Results” from the report selection window, this report can be viewed in the report viewing window. The user can zoom in and out of the report and print it using the “Print” button.

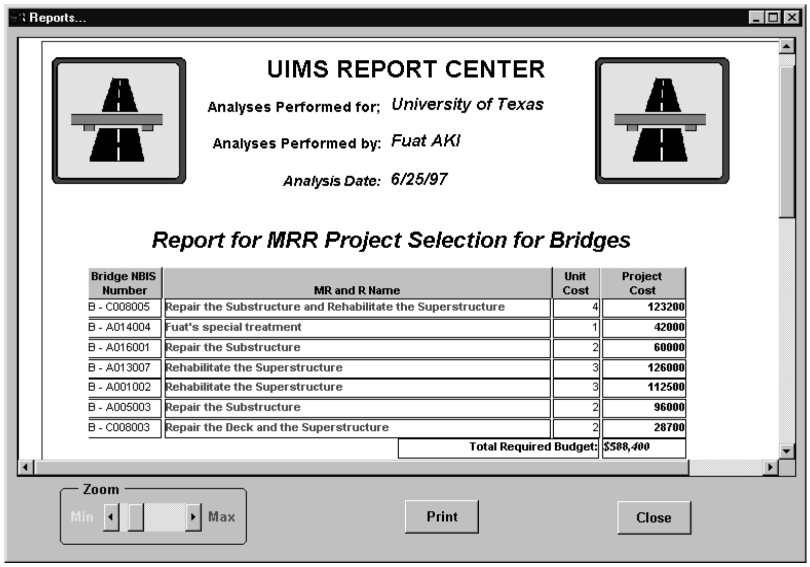


Figure 6.33 Report for Bridge MRR Results

CHAPTER 7
CONCLUSIONS AND RECOMMENDATIONS

An integrated urban infrastructure management system (UIMS) for managing urban transportation infrastructure was developed in this study. UIMS is a computerized tool which is intended to assist decision makers in managing their infrastructure effectively. The system consists of two subsystems: pavement management and bridge management. UIMS integrates these two separate systems by means of ultimate data sharing and also provides a single evaluation index to compare pavements and bridges simultaneously.

UIMS basically works in two ways: the user can use it either as a tool to manage pavements or bridges separately (like two separate programs), or make an integrated analysis for a more cost effective budget allocation and M&R project assignment. The UIMS software is a Windows-based application. It is a very user-friendly program supported with graphical user interface (GUI) technology.

UIMS is a very flexible program which makes it easier to adapt to different local conditions and requirements according to different agencies' needs. Most of the models used in the analyses are generic, and can be changed or modified by the user. UIMS filtering and sorting functions allow the user to set any kind of expressions when selecting or prioritizing M & R candidates. In other words, it does not restrict the user to a pre-defined priority index or a structural rating number.

The UIMS database is an object oriented database which contains inventory and condition data for both pavements and bridges. The software provides import and export options that make it faster in cases where data is available from external sources. UIMS can import text (ASCII) and Dbase (DBF) files which can be generated by most of the other computerized systems. The export option provides more alternatives to save the data into a specific file format which allows the user to make some other analyses using some other packages such as spreadsheet files like Excel. The main purpose of exporting is to provide a backup mechanism for the system.

The data items used in the bridge management part of the UIMS are NBIS data items. This makes it easier to implement UIMS compared to other bridge management tools, because one of the major problems in implementing a bridge management system, as stated in Chapter 3, is the compatibility with the NBIS database. The import option of the UIMS provides for the direct reading of data from the NBIS database, thus reducing the time of data entry.

UIMS is an effective tool in network level infrastructure management. It first provides some basic statistical values for the variables that are used in the analyses, including ADT, length, width, average bridge condition, and pavement condition index (PCI). These statistics are used to evaluate the pavement and bridge networks in general. The user then decides on the levels of these variables and sets the decision trees that will be used in selecting M & R actions. Next, UIMS applies these decision trees and assigns M & R projects to pavements and bridges. The cost of each project is calculated and compared with a user defined budget.

There are three different analysis options for bridges in UIMS: a flagging procedure that pops out the bridges with certain deficiencies such as scour critical and historical bridges, a decision tree procedure that assigns M & R projects to bridges, and another decision tree that assigns operational restrictions and/or special inspections. The special inspections procedure assigns an inspection date for the bridges that require inspection due to an operational deficiency.

The street map function in UIMS provides the user to view the pavements and bridges on the screen. This function provides a quick reference for the user to identify the structures according to their different attributes, such as low condition and high traffic. UIMS uses color coding technology for this purpose.

UIMS provides many different alternatives for reporting purposes. The pre-defined reports that are accessed through the reporting module are readily available for user to review and print reports. In addition, each window in the UIMS has a printing option that makes it possible to produce any kind of printouts that may be called as user defined reports.

The system developed in this study is a working system and can be implemented and used by public agencies to manage their pavement and bridge infrastructure. However, UIMS can be expanded into a more effective system through additional study:

1. UIMS should be tested using field data. The data used in developing UIMS approximates real data. In order to understand the effectiveness of the new system it should be tested with a real data set and the results should be compared with the results of other working systems prior to any implementation work.

2. A management system cannot provide any benefits to the public unless it is implemented. For this reason, the next step in the study should be the implementation of the UIMS. To implement the UIMS, several pilot cities should be selected first. According to the results of these pilot implementations, the scope can be broadened in the implementation.
3. Currently UIMS provides the user a one-year (current condition) analysis. A complete management system should include multi-year analyses options, including multi-year budget allocations and planning, performance modeling and future condition predictions.
4. UIMS is an effective tool for network level management. It addresses M & R needs, assigns specific projects for pavements and bridges and makes budget allocations between these projects. However it does not give the specific details for the selected projects. It would be better for the user to see the project level details so that a more cost effective design and construction alternatives can be generated. However, keep in mind that such a function would increase the data requirements.
5. GIS technology is becoming more and more popular, and it is obvious from the research studies that this technology can be applied to infrastructure management systems, providing substantial benefits to decision makers [Zhang 96]. The street map function of UIMS can be improved to include a complete GIS platform.

7.1 RECOMMENDATIONS FOR FUTURE DIRECTIONS

In order for the developed system to be complete and effective some modifications should be made. The most important ones can be categorized into two groups:

1. **Adding New Subsystems:** UIMS currently includes two subsystems: pavements and bridges. Although pavements and bridges are the most important types of transportation infrastructure, considering the fact that transportation infrastructure interacts closely with other types of infrastructure, a number of subsystems should be included in the integrated system. For example, water and wastewater lines and all other utility lines are placed under the pavements. In order to increase the

effectiveness of infrastructure management, these should be added to the integrated system. These infrastructure types are the following:

1) **Traffic signs:** The maintenance and replacement of traffic signs is an important issue that requires quite a bit of money. Similar to pavements and bridges, traffic signs should also be inventoried, inspected, and evaluated. After this evaluation, a proper maintenance or replacement activity can be assigned to these traffic signs which requires scheduling a maintenance program including the manpower and maintenance equipment. Including this infrastructure type within the integrated system can be beneficial in terms of assessing the labor and equipment needs.
2) **Sidewalks:** Sidewalks are an important type of infrastructure in urban areas where there is high pedestrian traffic. They always interact with pavements, and in most cases with bridges. They should be inventoried and inspected as well for an effective maintenance and rehabilitation program.
3) **Parking lots:** Parking lots are important types of infrastructure in cities especially where there are parking problems. Systematic procedures of maintaining and rehabilitating this infrastructure type may be very beneficial in reducing delays and congestion caused by vehicles parked along roadsides. Coordinating these with pavement and bridge management systems may be more beneficial in this respect.
4) **Drainage systems:** Although drainage systems are not considered types of transportation infrastructure, their inspection and maintenance are performed by roadway maintenance crews. In addition, these structures have a direct relationship with the roads such that their malfunctioning may increase the deterioration rate of pavements. For these reasons these structures should also be included in the integrated system.

2. **Adding New Functions:** The current system uses decision trees and prioritization techniques in analyses procedures. For more effective analyses and comparisons of different types of infrastructure, new functions can be developed. Among these, optimization functions based on life-cycle costs can provide more accurate results. Another development can be the addition of new procedures like knowledge based

systems that integrate the engineer's experience into the infrastructure management. In infrastructure management, the maintenance and rehabilitation procedures are based primarily on agency policies. Knowledge-based systems can be used to store the knowledge of experienced engineers for further reference by the non-experienced engineers.

REFERENCES

[Boyce 87] Boyce, C., Hudson, W.R., and Burns, N.H., "Improvements in On-System Bridge Project Prioritization," Research Report 439-1, Center for Transportation Research, The University of Texas at Austin, January 1987.

[Bridgit 94] *Bridgit User's Manual*, Prepared for National Cooperative Highway Research Program (NCHRP Project 12-28(2)A), December 1994.

[Brinsap 94] "Brinsap Manual of Procedures," *Coding Guide*, Prepared for Texas Department of Transportation, May 1994.

[Chen 93] Chen, X., "Development of an Urban Roadway Management System," Dissertation for Doctor of Philosophy, The University of Texas at Austin, December 1993.

[FHWA 89] Federal Highway Administration, Report No: FHWA-DP-71-01R, "Bridge Management System," October, 1989.

[Hearn 93] Hearn, G., Frangopol, D., Chakravorty, M., Myers, S., Pinkerton, B., and Siccardi, A.J., "Automated Generation of NBI Reporting Fields from Pontis BMS Database," *Infrastructure Planning and Management*, ASCE, 1993, pp. 226-230.

[Hudson 96] Hudson, W.R., Haas, R., Uddin, W., *Infrastructure Management*, McGraw-Hill, 1997.

[Hudson 94] Hudson, W.R., Haas, R., Zaniewski, J., *Modern Pavement Management*, Kreiger Publishing Company, Florida, 1994.

[Hudson 87] Hudson, S.W., Carmichael III, R.F., Moser, L.O., Hudson, W.R., and Wilkes, W.J., "Bridge Management Systems," National Cooperative Highway Research Program, Report 300, December 1987.

[Pontis 91] *Pontis User's Manual*, Prepared for the Federal Highway Administration, Pre-release Copy, Cambridge Systematics, Inc. & Optima Inc., December 1991.

[Power 96] *Power Builder Desktop Version 5.0 User's Manual,* "Getting Started, Project Primer, Connecting to Your Database," Sybase, Inc., 1996.

[Sachs 93] Sachs, P. and Smith, R., "How to Get Local Public Works Agencies to Use Infrastructure Management Approaches," *Infrastructure Planning and Management*, ASCE, 1993, pp. 26-30.

[Shahin 94] Shahin, M.Y., *Pavement Management for Airports, Roads, and Parking Lots*, Chapman & Hall, New York, N.Y., 1994.

[Siccardi 93] Siccardi, A.J. and Montoya, P.J., "Managing a Bridge Program for Scour," *Infrastructure Planning and Management*, ASCE, 1993, pp. 267-271.

[Sohail 96] Sohail, F., Dossey, T., and Hudson, W.R., "Implementation of the Urban Roadway Management System," Research Report: SWUTC/96/465550-1, Southwest University Transportation Center, CTR, The University of Texas at Austin, April 1996.

[TRDI 96] Visual/BMS, "Bridge Management System Description," Texas Research & Development Incorporated (TRDI), May 1996.

[Weissmann 90] Weissmann, J., "A Bridge Management System Module for the Selection of Rehabilitation and Replacement Projects," Dissertation for Doctor of Philosophy, The University of Texas at Austin, May 1990.

[Wells 93] Wells, D.T., Scherer, W.T., and Gomez, J.P., "The State of the Art Bridge Management Systems," *Infrastructure Planning and Management*, ASCE, 1993, pp. 182-186.

[Windows 95] Windows 95 Online Help, Microsoft Corporation, Copyright 1983-1995.

[Zhang 96] Zhang, Z., "A GIS Based and Multimedia Integrated Infrastructure Management System," Dissertation for Doctor of Philosophy, The University of Texas at Austin, August 1996.

[Zhang 94] Zhang, Z., Dossey, T., Weissmann, J. and Hudson, W.R., "GIS Integrated Pavement and Infrastructure Management in Urban Areas," *Transportation Research Record 1429*, Transportation Research Board, National Research Council, Washington DC., 1994, pp. 84-89.

APPENDIX A

DATABASE TABLES

Table A1 Pavement Inventory Table

Definition: Includes the data items related to the inventory data for pavements, basically management section definition, geometric properties, traffic, etc.

Column Name	Data Type	Null	Length	Key
section_length	numeric	N	7	N
pavement_type	char	N	1	N
location_to	char	N	50	N
location_from	char	N	50	N
street_name	char	N	60	N
section_code	char	N	15	Y
city_name	char	Y	20	N
y2	numeric	Y	4	N
x2	numeric	Y	4	N
y1	numeric	Y	4	N
x1	numeric	Y	4	N
soil_type	char	Y	20	N
condition_index	numeric	N	5	N
truck_percentage	numeric	Y	3	N
adt_growth_rate	numeric	Y	3	N
adt	numeric	N	5	N
capacity	numeric	Y	5	N
construction_year	date	N	4	N
number_of_lanes	integer	Y	4	N
pavement_width	numeric	N	4	N

Table A2 Pavement Distress Table

Definition: Includes the data items related to distress data for pavements. It is possible to put 10 different distress types for each pavement type (Flexible or Rigid).				
Column Name	**Data Type**	**Null**	**Length**	**Key**
fdist2	numeric	Y	5	N
fdist10	numeric	Y	5	N
rdist2	numeric	Y	5	N
rdist1	numeric	Y	5	N
fdist4	numeric	Y	5	N
section_code	char	N	15	Y
fdist3	numeric	Y	5	N
pavement_type	char	Y	1	N
fdist9	numeric	Y	5	N
rdist10	numeric	Y	5	N
rdist9	numeric	Y	5	N
fdist8	numeric	Y	5	N
fdist7	numeric	Y	5	N
fdist1	numeric	Y	5	N
rdist8	numeric	Y	5	N
rdist7	numeric	Y	5	N
rdist6	numeric	Y	5	N
rdist5	numeric	Y	5	N
fdist6	numeric	Y	5	N
rdist4	numeric	Y	5	N
rdist3	numeric	Y	5	N
fdist5	numeric	Y	5	N

Table A3 Pavement Distress Weights Table

Definition: This is a user-defined table (not an inventory table), but since the values are inputted directly by the user, it is still a row data table. It includes the data items related to distress weights for pavements for a possible distress index definition in the future. Weighting factors for each distress type for each severity level are stored in this table.

Column Name	Data Type	Null	Length	Key
pavement_type	char	Y	1	N
low_weight	numeric	N	4	N
distress_name	char	N	50	Y
med_weight	numeric	N	4	N
hig_weight	numeric	N	4	N

Table A4 Pavement MRR Action Definition Table

Definition: This is a user-defined table (not an inventory table), but since the values are inputted directly by the user, it is still a row data table. It includes the possible MRR actions that can be applied to pavements and their unit costs.

Column Name	Data Type	Null	Length	Key
pavement_type	char	Y	1	N
unit_cost	numeric	N	5	N
mrr_name	char	N	60	N
mrr_code	numeric	N	3	Y

Table A5 Pavement Decision Tree Structure Definition Table

Definition: This table includes the data columns that identify the decision tree structure for pavements.

Column Name	Data Type	Null	Length	Key
mrr19	numeric	N	3	N
mrr18	numeric	N	3	N
mrr17	numeric	N	3	N
mrr16	numeric	N	3	N
mrr15	numeric	N	3	N
mrr14	numeric	N	3	N
mrr13	numeric	N	3	N
mrr12	numeric	N	3	N
mrr11	numeric	N	3	N

Table A5 Pavement Decision Tree Structure Definition Table (Cont.)

mrr10	numeric	N	3	N
mrr9	numeric	N	3	N
mrr8	numeric	N	3	N
mrr7	numeric	N	3	N
mrr6	numeric	N	3	N
mrr5	numeric	N	3	N
mrr4	numeric	N	3	N
mrr3	numeric	N	3	N
mrr2	numeric	N	3	N
mrr1	numeric	N	3	N
key	char	N	1	Y
mrr27	numeric	N	5	N
mrr26	numeric	N	3	N
mrr25	numeric	N	3	N
mrr24	numeric	N	3	N
mrr23	numeric	N	3	N
mrr22	numeric	N	3	N
mrr21	numeric	N	3	N
mrr20	numeric	N	3	N

Table A6 Pavement Evaluation Definition Table

Definition: This table includes data columns that identify the five levels for PCI, ADT, and AGE for pavements.

Column Name	Data Type	Null	Length	Key
traf4	numeric	N	7	N
traf3	numeric	N	7	N
traf2	numeric	N	7	N
traf1	numeric	N	7	N
age4	integer	N	4	N
age3	integer	N	4	N
age2	integer	N	4	N
age1	integer	N	4	N
pci4	integer	N	4	N
pci3	integer	N	4	N
pci2	integer	N	4	N
pci1	integer	N	4	Y

Table A7 Bridge Inventory Table

Definition: Includes the data items related to the inventory data for bridges.

truck_percentage	numeric	Y	3	N
city_name	char	Y	20	N
deck_width	numeric	N	5	N

bridge_construction_type	char	Y	30	N
historical_significance	integer	N	4	N
x_coordinate	numeric	Y	4	N
adt_growth_rate	numeric	Y	3	N
y_coordinate	numeric	Y	4	N
minimum_under_clearance	char	Y	6	N
traffic_safety_features	char	N	1	N
inspection_date	date	Y	4	N
bridge_nbis_number	char	N	15	Y
side_walk_width	numeric	Y	5	N
adt	numeric	N	5	N
construction_date	date	N	4	N
bridge_material_type	char	Y	30	N
number_of_lanes	integer	Y	4	N
bridge_open_code	char	Y	1	N
number_of_spans	integer	Y	4	N
scour_critical_code	integer	N	4	N
load_restriction	char	N	1	N
design_loading	char	N	20	N
bridge_location	char	N	60	N
street_name	char	N	60	N
bridge_length	numeric	N	6	N

Table A8 Bridge Condition Table

Definition: Includes the data items related to the condition of bridges.

Column Name	Data Type	Null	Length	Key
bridge_nbis_number	char	N	15	Y
deck_condition	numeric	N	4	N
substructure_condition	numeric	N	4	N
superstructure_condition	numeric	N	4	N

Table A9 Bridge MRR Action Definition Table

Definition: This is a user defined table (not an inventory table), but since the values are inputted directly by the user, it is still a row data table. It includes the possible MRR actions that can be applied to the bridges and their unit costs.

Column Name	Data Type	Null	Length	Key
unit	char	Y	10	N
unit_cost	numeric	N	5	N
mrr_name	char	N	60	N
mrr_code	numeric	N	3	Y

Table A10 Bridge Inspection Action Definition Table

Definition: This table includes the possible inspection actions that can be applied to the bridges.				
Column Name	**Data Type**	**Null**	**Length**	**Key**
inspection_action	char	N	85	N
inspection_code	numeric	N	3	Y
inspection_frequency	numeric	Y	2	N
inspection_cost	numeric	Y	5	N

Table A11 Flagging Filter Definition Table for Bridges

Definition: This table includes the definition of critical values for the variables that are used in the flagging procedure for bridges.				
Column Name	**Data Type**	**Null**	**Length**	**Key**
load_restriction	char	N	20	N
deck_width	numeric	N	7	N
scour_critical	numeric	N	4	N
safety_features	numeric	N	3	N
condition	numeric	N	4	N
adt	numeric	N	5	N
bridge_length	numeric	N	6	N
historical_significance	integer	N	4	Y

Table A12 Decision Tree Structure Definition for MRR Actions for Bridges

Definition: This is a user defined table (not an inventory table), but since the values are inputted directly by the user, it is still a row data table. It includes the definition of the decision tree that will be applied to select the MRR projects for bridges.				
Column Name	**Data Type**	**Null**	**Length**	**Key**
key	char	N	1	Y
mrr27	numeric	N	5	N
mrr26	numeric	N	3	N
mrr25	numeric	N	3	N
mrr24	numeric	N	3	N
mrr23	numeric	N	3	N
mrr22	numeric	N	3	N
mrr21	numeric	N	3	N
mrr20	numeric	N	3	N
mrr19	numeric	N	3	N
mrr18	numeric	N	3	N
mrr17	numeric	N	3	N
mrr16	numeric	N	3	N
mrr15	numeric	N	3	N
mrr14	numeric	N	3	N
mrr13	numeric	N	3	N
mrr12	numeric	N	3	N
mrr11	numeric	N	3	N
mrr10	numeric	N	3	N
mrr9	numeric	N	3	N
mrr8	numeric	N	3	N
mrr7	numeric	N	3	N
mrr6	numeric	N	3	N
mrr5	numeric	N	3	N
mrr4	numeric	N	3	N
mrr3	numeric	N	3	N
mrr2	numeric	N	3	N
mrr1	numeric	N	3	N

Table A13 Decision Tree Structure Definition for Inspections for Bridges

Definition: This is a user defined table (not an inventory table), but since the values are inputted directly by the user it is still a row data table. It includes the definition of the decision tree that will be applied to select the special inspection activities for bridges.				
Column Name	**Data Type**	**Null**	**Length**	**Key**
insp15	numeric	N	3	N
insp9	numeric	N	3	N
insp19	numeric	N	3	N
insp11	numeric	N	3	N
insp18	numeric	N	3	N
key	char	N	1	Y
insp17	numeric	N	3	N
insp10	numeric	N	3	N
insp14	numeric	N	3	N
insp2	numeric	N	3	N
insp6	numeric	N	3	N
insp5	numeric	N	3	N
insp20	numeric	N	3	N
insp4	numeric	N	3	N
insp3	numeric	N	3	N
insp1	numeric	N	3	N
insp7	numeric	N	3	N
insp16	numeric	N	3	N
insp8	numeric	N	3	N
insp24	numeric	N	3	N
insp23	numeric	N	3	N
insp12	numeric	N	3	N
insp22	numeric	N	3	N
insp13	numeric	N	3	N
insp21	numeric	N	3	N

Table A14 Decision Tree Variables Definition Table

Definition: This table includes data columns that identify the three levels for deck, substructure and superstructure conditions for bridges.

Column Name	Data Type	Null	Length	Key
superstructure_2	numeric	N	4	N
superstructure_1	numeric	N	4	N
substructure_2	numeric	N	4	N
substructure_1	numeric	N	4	N
deck_2	numeric	N	4	N
deck_1	numeric	N	4	Y

Table A15 Decision Tree Variables Definition Table

Definition: This table includes data columns that identify the levels for width, safety features, load restriction, and scour critical variables for bridges.

Column Name	Data Type	Null	Length	Key
width	numeric	N	6	Y
safety_features	numeric	N	3	N
load_restriction	char	N	20	N
scour_very_critical	numeric	N	3	N
scour_critical	numeric	N	3	N

Table A16 User Information Table

Definition: This table includes data columns that contain the information related to identifying the system user.

Column Name	Data Type	Null	Length	Key
sign_on_date	date	Y	4	N
agency_name	char	N	20	N
user_name	char	N	20	Y

Table A17 Budget Table

Definition: This table includes data columns that contain the budget information.

Column Name	Data Type	Null	Length	Key
agency_name	numeric	Y	12	N
user_name	numeric	Y	12	N
key	char	N	1	Y

Table A18 Pavement Results Table

Column Name	Data Type	Null	Length	Key
Definition: This table is used in pavement management analyses. It is a combination of pavement inventory, pavement evaluation, pavement MRR definition, and pavement decision tree structure definition tables.				
location_to	char	N	50	N
number_of_lanes	integer	Y	4	N
section_length	numeric	N	7	N
adt_growth_rate	numeric	Y	3	N
truck_percentage	numeric	Y	3	N
capacity	numeric	Y	5	N
condition_index	numeric	N	5	N
age	numeric	Y	5	N
adt_low	numeric	Y	7	N
age_low	integer	Y	4	N
age_up	integer	Y	4	N
mrrp27	numeric	Y	3	N
mrrp23	numeric	Y	3	N
mrrp21	numeric	Y	3	N
mrrp20	numeric	Y	3	N
mrrp19	numeric	Y	3	N
mrrp15	numeric	Y	3	N
mrrp13	numeric	Y	3	N
mrrp11	numeric	Y	3	N
mrrp7	numeric	Y	3	N
mrrp5	numeric	Y	3	N
mrrp3	numeric	Y	3	N
y2	numeric	Y	4	N
y1	numeric	Y	4	N
x1	numeric	Y	4	N
soil_type	char	Y	20	N
adt	numeric	N	5	N
construction_year	date	N	4	N
pavement_width	numeric	N	4	N
pavement_type	char	N	1	N
location_from	char	N	50	N
street_name	char	N	60	N
section_code	char	N	15	Y

Table A18 Pavement Results Table (Cont.)

mrrp9	numeric	Y	3	N
mrrp12	numeric	Y	3	N
mrrp10	numeric	Y	3	N
mrrp8	numeric	Y	3	N
mrrp18	numeric	Y	3	N
mrrp17	numeric	Y	3	N
mrrp16	numeric	Y	3	N
mrrp22	numeric	Y	3	N
x2	numeric	Y	4	N
mrrp26	numeric	Y	3	N
mrrp25	numeric	Y	3	N
mrrp24	numeric	Y	3	N
mrrp14	numeric	Y	3	N
unit_cost	numeric	Y	5	N
pci_low	integer	Y	4	N
pci_up	integer	Y	4	N
mrrp1	numeric	Y	3	N
mrrp4	numeric	Y	3	N
mrrp6	numeric	Y	3	N
adt_up	numeric	Y	7	N

Table A19 Bridge Network Analysis Table

Definition: This table contains the flagging filter information for bridges. The contents of this table are the bridge inventory data that are filtered according to the flagging filtering criteria. For this reason, the columns of this table are the same columns as the bridge inventory table. This table is used for two purposes: to pop out (flag) the bridges with certain deficiency, such as scour, width, etc.; and, to provide a network level filter for the bridge population to reduce the number of bridges to be analyzed.				
Column Name	**Data Type**	**Null**	**Length**	**Key**
Same columns in bridge inventory table.				

Table A20 Bridge MRR Results Table

Definition: This table contains the information related to bridge MRR project selection. It contains parts of the bridge inventory data table, bridge decision tree structure definition table (for MRR), decision tree variables definition table, and MRR actions definition table.				
Column Name	**Data Type**	**Null**	**Length**	**Key**
mrrc14	numeric	Y	3	N
mrrc15	numeric	Y	3	N
mrrc24	numeric	Y	3	N
mrrc9	numeric	Y	3	N
mrrc25	numeric	Y	3	N
mrrc23	numeric	Y	3	N
mrrc1	numeric	Y	3	N
mrrc27	numeric	Y	3	N
substructure_1	numeric	Y	4	N
mrrc22	numeric	Y	3	N
mrrc6	numeric	Y	3	N
mrrc20	numeric	Y	3	N
mrrc5	numeric	Y	3	N
mrrc8	numeric	Y	3	N
mrrc4	numeric	Y	3	N
mrrc12	numeric	Y	3	N
mrrc10	numeric	Y	3	N
mrrc2	numeric	Y	3	N
deck_2	numeric	Y	4	N
superstructure_condition	numeric	Y	4	N
mrrc26	numeric	Y	3	N
deck_1	numeric	Y	4	N
mrrc13	numeric	Y	3	N
mrr_code	integer	Y	4	N
substructure_2	numeric	Y	4	N
deck_condition	numeric	Y	4	N
mrrc21	numeric	Y	3	N
total_cost	numeric	Y	7	N
mrrc19	numeric	Y	3	N
bridge_nbis_number	char	N	15	N
mrrc18	numeric	Y	3	N
unit_cost	numeric	Y	5	N

Table A20 Bridge MRR Results Table (Cont.)

mrrc11	numeric	Y	3	N
mrrc3	numeric	Y	3	N
superstructure_2	numeric	Y	4	N
mrrc7	numeric	Y	3	N
superstructure_1	numeric	Y	4	N
substructure_condition	numeric	Y	4	N
mrrc16	numeric	Y	3	N
mrrc17	numeric	Y	3	N

Table A21 Bridge Inspection Results Table

Definition: This table contains the information related to special inspection activity selection for bridges. It consists of some part of the bridge inventory data table, bridge decision tree structure definition table (for Special Inspections), decision tree variables definition table, and Inspection actions definition table.

Column Name	Data Type	Null	Length	Key
inspc1	numeric	Y	3	N
inspc2	numeric	Y	3	N
inspc19	numeric	Y	3	N
inspection_code	numeric	N	3	N
inspc5	numeric	Y	3	N
inspc22	numeric	Y	3	N
deck_width	numeric	Y	5	N
scour_code	integer	Y	4	N
inspc6	numeric	Y	3	N
inspc20	numeric	Y	3	N
inspc11	numeric	Y	3	N
inspc10	numeric	Y	3	N
inspc13	numeric	Y	3	N
inspc15	numeric	Y	3	N
inspc3	numeric	Y	3	N
bridge_nbis_number	char	N	15	N
inspc23	numeric	Y	3	N
safety	char	Y	1	N
loading	char	Y	20	N

Table A21 Bridge Inspection Results Table (Cont.)

inspc16	numeric	Y	3	N
scour_very_critical	numeric	N	3	N
inspc24	Numeric	Y	3	
inspc14	numeric	Y	3	N
inspection_cost	numeric	N	5	N
inspc8	numeric	Y	3	N
inspc9	numeric	Y	3	N
inspc4	numeric	Y	3	N
inspc17	numeric	Y	3	N
safety_features	numeric	N	3	N
inspc12	numeric	Y	3	N
inspc18	numeric	Y	3	N
inspc7	numeric	Y	3	N
inspection_frequency	numeric	N	2	N
load_restriction	char	N	20	N
scour_critical	numeric	N	3	N
width	numeric	N	6	N
inspc21	numeric	Y	3	N

APPENDIX B

EXAMPLE SOURCE CODES FOR UIMS SOFTWARE

B.1) Example Source Code for "Apply Decision Tree" Function

```
int row_coun
string section
dw_2.reset()
return_value=2
setpointer(HourGlass!)
delete from "pavement_results";
insert into "pavement_results" ("section_code",
        "street_name",
        "location_from",
        "location_to",
        "pavement_type",
        "section_length",
        "pavement_width",
        "number_of_lanes",
        "construction_year",
        "capacity",
        "adt",
        "adt_growth_rate",
        "truck_percentage",
        "condition_index",
        "soil_type",
        "x1",
        "y1",
        "x2",
        "y2",
        "city_name",
        "mrrp1",
        "mrrp2",
        "mrrp3",
        "mrrp4",
        "mrrp5",
        "mrrp6",
        "mrrp7",
        "mrrp8",
        "mrrp9",
        "mrrp10",
        "mrrp11",
        "mrrp12",
        "mrrp13",
        "mrrp14",
```

```
        "mrrp15",
        "mrrp16",
        "mrrp17",
        "mrrp18",
        "mrrp19",
        "mrrp20",
        "mrrp21",
        "mrrp22",
        "mrrp23",
        "mrrp24",
        "mrrp25",
        "mrrp26",
        "mrrp27",
        "age_up",
        "age_low",
        "adt_up",
        "adt_low",
        "pci_up",
        "pci_low",
        "unit_cost",
        "age")
select "pavement_network_analysis"."section_code",
        "pavement_network_analysis"."street_name",
        "pavement_network_analysis"."location_from",
        "pavement_network_analysis"."location_to",
        "pavement_network_analysis"."pavement_type",
        "pavement_network_analysis"."section_length",
        "pavement_network_analysis"."pavement_width",
        "pavement_network_analysis"."number_of_lanes",
        "pavement_network_analysis"."construction_year",
        "pavement_network_analysis"."capacity",
        "pavement_network_analysis"."adt",
        "pavement_network_analysis"."adt_growth_rate",
        "pavement_network_analysis"."truck_percentage",
        "pavement_network_analysis"."condition_index",
        "pavement_network_analysis"."soil_type",
        "pavement_network_analysis"."x1",
        "pavement_network_analysis"."y1",
        "pavement_network_analysis"."x2",
        "pavement_network_analysis"."y2",
        "pavement_network_analysis"."city_name",
        0,
        0,
        0,
        0,
```

```
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          0,
          1997-year("pavement_network_analysis"."construction_year")
from "pavement_network_analysis";

update "pavement_results"
          set "age_up" =
          (select "pavement_evaluation"."age3" from "pavement_evaluation");
update "pavement_results"
          set "age_low" =
          (select "pavement_evaluation"."age2" from "pavement_evaluation");
update "pavement_results"
          set "pci_up" =
          (select "pavement_evaluation"."pci3" from "pavement_evaluation");
update "pavement_results"
          set "pci_low" =
          (select "pavement_evaluation"."pci2" from "pavement_evaluation");
```

```
update "pavement_results"
        set "adt_up" =
        (select "pavement_evaluation"."traf3" from "pavement_evaluation");
update "pavement_results"
        set "adt_low" =
        (select "pavement_evaluation"."traf2" from "pavement_evaluation");
update "pavement_results"
        set "mrrp1" =
        (select "tree_structure_pavement_f"."mrr1" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" >="pavement_results"."age_low" and "pavement_results"."adt"
        >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp2" =
        (select "tree_structure_pavement_f"."mrr2" from "tree_structure_pavement_f" where
        "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp3" =
        (select "tree_structure_pavement_f"."mrr3" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp4" =
        (select "tree_structure_pavement_f"."mrr4" from     “trree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp5" =
        (select "tree_structure_pavement_f"."mrr5" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp6" =
        (select "tree_structure_pavement_f"."mrr6" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" <"pavement_results"."age_low" and
```

```
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp7" =
        (select "tree_structure_pavement_f"."mrr7" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp8" =
        (select "tree_structure_pavement_f"."mrr8" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp9" =
        (select "tree_structure_pavement_f"."mrr9" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" <"pavement_results"."pci_low" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp10" =
        (select "tree_structure_pavement_f"."mrr10" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp11" =
        (select "tree_structure_pavement_f"."mrr11" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp12" =
        (select "tree_structure_pavement_f"."mrr12" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp13" =
```

```
        (select "tree_structure_pavement_f"."mrr13" from "tree_structure_pavement_f" where
        "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp14" =
        (select "tree_structure_pavement_f"."mrr14" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp15" =
        (select "tree_structure_pavement_f"."mrr15" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
         "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp16" =
        (select "tree_structure_pavement_f"."mrr16" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp17" =
        (select "tree_structure_pavement_f"."mrr17" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp18" =
        (select "tree_structure_pavement_f"."mrr18" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_low" and
        "pavement_results"."condition_index" <"pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
```

```
update "pavement_results"
        set "mrrp19" =
        (select "tree_structure_pavement_f"."mrr19" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp20" =
        (select "tree_structure_pavement_f"."mrr20" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp21" =
        (select "tree_structure_pavement_f"."mrr21" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" >="pavement_results"."age_low" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp22" =
        (select "tree_structure_pavement_f"."mrr22" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp23" =
        (select "tree_structure_pavement_f"."mrr23" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp24" =
        (select "tree_structure_pavement_f"."mrr24" from "tree_structure_pavement_f"
         where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_low" and
        "pavement_results"."age" >="pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp25" =
        (select "tree_structure_pavement_f"."mrr25" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
```

```
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" >= "pavement_results"."adt_low") ;
update "pavement_results"
        set "mrrp26" =
        (select "tree_structure_pavement_f"."mrr26" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_low" and
        "pavement_results"."adt" >= "pavement_results"."adt_up") ;
update "pavement_results"
        set "mrrp27" =
        (select "tree_structure_pavement_f"."mrr27" from "tree_structure_pavement_f"
        where "pavement_results"."condition_index" >="pavement_results"."pci_up" and
        "pavement_results"."age" <"pavement_results"."age_up" and
        "pavement_results"."adt" < "pavement_results"."adt_up") ;
update "pavement_results"
        set "unit_cost" =
        (select "pavement_mrr_actions"."unit_cost" from "pavement_mrr_actions" where
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp1" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp2" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp3" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp4" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp5" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp6" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp7" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp8" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp9" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp10" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp11" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp12" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp13" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp14" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp15" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp16" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp17" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp18" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp19" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp20" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp21" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp22" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp23" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp24" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp25" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp26" or
        "pavement_mrr_actions"."mrr_code"="pavement_results"."mrrp27" );
```

```
commit;

row_coun=dw_1.rowcount()
curr_row=1
do while curr_row<=row_coun
	section=dw_1.getitemstring(curr_row,"section_code")
	dw_2.retrieve(section)
	curr_row=curr_row+1
loop

setpointer(Arrow!)
```

B.2) Example Source Code for "Prioritize" Function

```
setpointer(hourglass!)
delete from "prioritization";
insert into "prioritization" ("structure_no",
	"street_name",
	"condition",
	"adt",
	"width",
	"age",
	"condition_low",
	"condition_up",
	"adt_low",
	"adt_up",
	"width_low",
	"width_up",
	"age_low",
	"age_up",
	"pi1_1",
	"pi1_2",
	"pi1_3",
	"pi1_4",
	"pi1_5",
	"pi1_6",
	"pi1_7",
	"pi1_8",
	"pi1_9",
	"pi1_10",
	"pi1_11",
	"pi1_12",
	"pi1_13",
	"pi1_14",
	"pi1_15",
```

```
        "pi1_16",
        "pi1_17",
        "pi1_18",
        "pi1_19",
        "pi1_20",
        "pi1_21",
        "pi1_22",
        "pi1_23",
        "pi1_24",
        "pi1_25",
        "pi1_26",
        "pi1_27",
        "pi2_1",
        "pi2_2",
        "pi2_3",
        "pi2_4",
        "pi2_5",
        "pi2_6",
        "pi2_7",
        "pi2_8",
        "pi2_9",
        "pi2_10",
        "pi2_11",
        "pi2_12",
        "pi2_13",
        "pi2_14",
        "pi2_15",
        "pi2_16",
        "pi2_17",
        "pi2_18",
        "pi2_19",
        "pi2_20",
        "pi2_21",
        "pi2_22",
        "pi2_23",
        "pi2_24",
        "pi2_25",
        "pi2_26",
        "pi2_27",
        "pi")
select "pavement_results"."section_code",
        "pavement_results"."street_name",
        "pavement_results"."condition_index",
        "pavement_results"."adt",
        "pavement_results"."pavement_width",
```

```
        "pavement_results"."age",
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
from "pavement_results";
insert into "prioritization" ("structure_no",
        "street_name",
        "condition",
        "adt",
        "width",
        "age",
        "condition_low",
        "condition_up",
        "adt_low",
        "adt_up",
        "width_low",
        "width_up",
        "age_low",
        "age_up",
        "pi1_1",
        "pi1_2",
        "pi1_3",
        "pi1_4",
        "pi1_5",
        "pi1_6",
        "pi1_7",
        "pi1_8",
```

"pi1_9",
"pi1_10",
"pi1_11",
"pi1_12",
"pi1_13",
"pi1_14",
"pi1_15",
"pi1_16",
"pi1_17",
"pi1_18",
"pi1_19",
"pi1_20",
"pi1_21",
"pi1_22",
"pi1_23",
"pi1_24",
"pi1_25",
"pi1_26",
"pi1_27",
"pi2_1",
"pi2_2",
"pi2_3",
"pi2_4",
"pi2_5",
"pi2_6",
"pi2_7",
"pi2_8",
"pi2_9",
"pi2_10",
"pi2_11",
"pi2_12",
"pi2_13",
"pi2_14",
"pi2_15",
"pi2_16",
"pi2_17",
"pi2_18",
"pi2_19",
"pi2_20",
"pi2_21",
"pi2_22",
"pi2_23",
"pi2_24",
"pi2_25",
"pi2_26",

```
        "pi2_27",
        "pi")
 select "bridge_inventory"."bridge_nbis_number",
        "bridge_inventory"."street_name",
        (100/9)*("bridge_condition"."deck_condition"+"bridge_condition"."su
        bstructure_condition"+"bridge_condition"."superstructure_condition")    /3,
"bridge_inventory"."adt",
        "bridge_inventory"."deck_width",
        (1997-year("bridge_inventory"."construction_date")),
        0,
        0,
        0,
        0,
        0,
        0,
        0
from "bridge_inventory","bridge_condition"
        where
        “bridge_inventory"."bridge_nbis_number"="bridge_condition"."bridge_nbis_number";

update "prioritization"
        set "width_low" =
        (select (max("pavement_width")/3) from "pavement_results");
update "prioritization"
        set "width_up" =
        (select (2*max("pavement_width")/3) from "pavement_results");
update "prioritization"
        set "condition_low" =
        (select "pci2" from "pavement_evaluation");
update "prioritization"
        set "condition_up" =
        (select "pci4" from "pavement_evaluation");
update "prioritization"
        set "adt_low" =
        (select "traf3" from "pavement_evaluation");
update "prioritization"
        set "adt_up" =
        (select "traf1" from "pavement_evaluation");
update "prioritization"
        set "age_low" =
        (select "age4" from "pavement_evaluation");
update "prioritization"
        set "age_up" =
        (select "age2" from "pavement_evaluation");
update "prioritization"
```

```
        set pi1_1 = (select 1
        where (condition<condition_low and age>age_up))
update "prioritization"
        set pi1_2 = (select 2
        where (condition<condition_low and age>age_low and age<=age_up))
update "prioritization"
        set pi1_3 = (select 3
        where (condition<condition_low and age<=age_low))
update "prioritization"
        set pi1_4 = (select 4
        where (condition>=condition_low and condition<condition_up and age>age_up))
update "prioritization"
        set pi1_5 = (select 5
        where (condition>=condition_low and condition<condition_up and age>age_low and
        age<=age_up))
update "prioritization"
        set pi1_6 = (select 6
        where (condition>=condition_low and condition<condition_up and age<=age_low))
update "prioritization"
        set pi1_7 = (select 7
        where (condition>=condition_up and age>age_up))
update "prioritization"
        set pi1_8 = (select 8
        where (condition>=condition_up and age>age_low and age<=age_up))
update "prioritization"
        set pi1_9 = (select 9
        where (condition>=condition_up and age<=age_low))
update "prioritization"
        set pi2_1 = (select 1
        where (adt>adt_up and width<width_low))
update "prioritization"
        set pi2_2 = (select 2
        where (adt>adt_up and width>=width_low and width<width_up))
update "prioritization"
        set pi2_3 = (select 3
        where (adt>adt_up and width>=width_up))
update "prioritization"
        set pi2_4 = (select 4
        where (adt<=adt_up and adt>adt_low and width<width_low))
update "prioritization"
        set pi2_5 = (select 5
        where (adt<=adt_up and adt>adt_low and width>=width_low and width<width_up))
update "prioritization"
        set pi2_6 = (select 6
        where (adt<=adt_up and adt>adt_low and width>=width_up))
```

```
update "prioritization"
        set pi2_7 = (select 7
        where (adt<=adt_low and width<width_low))
update "prioritization"
        set pi2_8 = (select 8
        where (adt<=adt_low and width>=width_low and width<width_up))
update "prioritization"
        set pi2_9 = (select 9
        where (adt<=adt_low and width>=width_up)) ;
update "prioritization"
        set pi =
        (isnull("prioritization"."pi1_1", "prioritization"."pi1_1",
        0)+isnull("prioritization"."pi1_2", "prioritization"."pi1_2",
        0)+isnull("prioritization"."pi1_3", "prioritization"."pi1_3",
        0)+isnull("prioritization"."pi1_4", "prioritization"."pi1_4",
        0)+isnull("prioritization"."pi1_5", "prioritization"."pi1_5",
        0)+isnull("prioritization"."pi1_6", "prioritization"."pi1_6",
        0)+isnull("prioritization"."pi1_7", "prioritization"."pi1_7",
        0)+isnull("prioritization"."pi1_8", "prioritization"."pi1_8",
        0)+isnull("prioritization"."pi1_9", "prioritization"."pi1_9",
        0)+isnull("prioritization"."pi2_1", "prioritization"."pi2_1",
        0)+isnull("prioritization"."pi2_2", "prioritization"."pi2_2",
        0)+isnull("prioritization"."pi2_3", "prioritization"."pi2_3",
        0)+isnull("prioritization"."pi2_4", "prioritization"."pi2_4",
        0)+isnull("prioritization"."pi2_5", "prioritization"."pi2_5",
        0)+isnull("prioritization"."pi2_6", "prioritization"."pi2_6",
        0)+isnull("prioritization"."pi2_7", "prioritization"."pi2_7",
        0)+isnull("prioritization"."pi2_8", "prioritization"."pi2_8",
        0)+isnull("prioritization"."pi2_9", "prioritization"."pi2_9", 0));
dw_3.Retrieve()
setpointer(arrow!)
```

B.3) Example Source Code for "Calculate" Function

```
int budget,cost,tot_cost,A,row_coun
dw_3.update()
row_coun=dw_2.rowcount()
A=1
do while A<=row_coun
dw_2.SelectRow(A, false)
A=A+1
loop
budget=dw_3.getitemnumber(1,"pavement_budget")
cost=0
curr_row=1
```

```
tot_cost=dw_2.getitemnumber(1,"compute_0008")
do while tot_cost <= budget
dw_2.SelectRow(curr_row, TRUE)
if curr_row=row_coun then EXIT
tot_cost=tot_cost+cost
cost=dw_2.getitemnumber(curr_row,"compute_0008")
curr_row=curr_row+1
loop
```

B.4) Example Source Code for "Sort" Function

```
int row_coun
string null_str
string section
dw_2.reset()
return_value=2
SetNull(null_str)
dw_1.setsort(null_str)
dw_1.sort()
row_coun=dw_1.rowcount()
curr_row=1
do while curr_row<=row_coun
        section=dw_1.getitemstring(curr_row,"section_code")
        dw_2.retrieve(section)
        curr_row=curr_row+1
loop
```

B.5) Example Source Code for "Filter" Function

```
int row_coun
string null_str
string section
dw_2.reset()
return_value=2
SetNull(null_str)
dw_1.setfilter(null_str)
dw_1.filter()
row_coun=dw_1.rowcount()
curr_row=1
do while curr_row<=row_coun
        section=dw_1.getitemstring(curr_row,"section_code")
        dw_2.retrieve(section)
        curr_row=curr_row+1
loop
```